# UNIVERSAL ROBOTS

## UR-6-85-5-A

## User Manual

Version 1.11, January 2010

# Contents


**1 Getting started** **7**
- 1.1 Introduction . . . 7
  - 1.1.1 The Robot . . . 8
  - 1.1.2 Programs . . . 8
  - 1.1.3 Safety Evaluation . . . 9
- 1.2 Turning On and Off . . . 9
  - 1.2.1 Turning on the Controller Box . . . 9
  - 1.2.2 Turning on the Robot . . . 9
  - 1.2.3 Initializing the Robot . . . 9
  - 1.2.4 Shutting Down the Robot . . . 9
  - 1.2.5 Shutting Down the Controller Box . . . 10
- 1.3 Quick start, Step by Step . . . 10
- 1.4 Mounting Instructions . . . 12
  - 1.4.1 The Workspace of the Robot . . . 12
  - 1.4.2 Mounting the Robot . . . 12
  - 1.4.3 Mounting the Tool . . . 12
  - 1.4.4 Mounting the Controller Box . . . 15
  - 1.4.5 Mounting the Touch Panel . . . 15
  - 1.4.6 Connecting the Robot Cable . . . 15
  - 1.4.7 Connecting the Mains Cable . . . 15

**2 Electrical Interface** **17**
- 2.1 Introduction . . . 17
- 2.2 The Emergency Stop Interface . . . 18
  - 2.2.1 The Simplest Emergency Stop Configuration . . . 19
  - 2.2.2 Connecting an External Emergency Stop Button . . . 20
  - 2.2.3 Using an External Emergency Stop Power Supply . . . 20
  - 2.2.4 Connecting to Other Machinery . . . 20
- 2.3 The Pause Interface . . . 21
  - 2.3.1 Connecting to the Pause Interface . . . 21
- 2.4 Controller I/O . . . 22
  - 2.4.1 Digital Outputs . . . 23
  - 2.4.2 Digital Inputs . . . 24
  - 2.4.3 Analog Outputs . . . 26
  - 2.4.4 Analog Inputs . . . 27
- 2.5 Tool I/O . . . 28
  - 2.5.1 Digital Outputs . . . 29
  - 2.5.2 Digital Inputs . . . 30
  - 2.5.3 Analog Inputs . . . 31

**3 PolyScope Software 33**
3.1 Introduction . . . 34
3.1.1 Welcome Screen . . . 35
3.1.2 Initialization Screen . . . 35
3.2 On-screen Editors . . . 37
3.2.1 On-screen Keypad . . . 37
3.2.2 On-screen Keyboard . . . 37
3.2.3 On-screen Expression Editor . . . 38
3.3 Robot Control . . . 38
3.3.1 Move Tab . . . 38
3.3.2 I/O Tab . . . 40
3.3.3 AutoMove Tab . . . 41
3.3.4 Setup → Mounting . . . 42
3.3.5 Setup → TCP Position . . . 43
3.3.6 Log Tab . . . 44
3.3.7 Load Screen . . . 44
3.3.8 Run Tab . . . 46
3.4 Programming . . . 46
3.4.1 Program → New Program . . . 47
3.4.2 Program Tab . . . 47
3.4.3 Program → Command Tab, <Empty> . . . 48
3.4.4 Program → Command Tab, Move . . . 49
3.4.5 Program → Command Tab, Fixed Waypoint . . . 50
3.4.6 Program → Command Tab, Relative Waypoint . . . 51
3.4.7 Program → Command Tab, Variable Waypoint . . . 52
3.4.8 Program → Command Tab, Wait . . . 53
3.4.9 Program → Command Tab, Action . . . 53
3.4.10 Program → Command Tab, Popup . . . 54
3.4.11 Program → Command Tab, Halt . . . 54
3.4.12 Program → Command Tab, Comment . . . 55
3.4.13 Program → Command Tab, Folder . . . 55
3.4.14 Program → Command Tab, Loop . . . 56
3.4.15 Program → Command Tab, SubProgram . . . 56
3.4.16 Program → Command Tab, Assignment . . . 57
3.4.17 Program → Command Tab, If . . . 58
3.4.18 Program → Command Tab, Script . . . 59
3.4.19 Program → Command Tab, Event . . . 59
3.4.20 Program → Command Tab, Thread . . . 60
3.4.21 Program → Command Tab, Pattern . . . 60
3.4.22 Program → Command Tab, Pallet . . . 62
3.4.23 Program → Command Tab, Stack . . . 63
3.4.24 Program → Graphics Tab . . . 67
3.4.25 Program → Structure Tab . . . 68
3.5 Setup . . . 69
3.5.1 Setup Screen . . . 69
3.5.2 Setup Screen → Initialize . . . 70
3.5.3 Setup Screen → Update . . . 70
3.5.4 Setup Screen → Password . . . 71
3.5.5 Setup Screen → Calibrate Touch Screen . . . 71
3.5.6 Setup Screen → Network . . . 72

**4 Warranties and Declarations** **73**
4.1 Warranty . . . 73
4.1.1 Product Warranty . . . 73
4.1.2 Disclaimer . . . 73
4.2 Declaration of Incorporation . . . 74
4.2.1 Product manufacturer . . . 74
4.2.2 Person Authorised to Compile the Technical Documentation 74
4.2.3 Description and Identification of Product . . . 74
4.2.4 Essential Requirements . . . 74
4.2.5 National Authority Contact Information . . . 75
4.2.6 Important Notice . . . 76
4.2.7 Place and Date of the Declaration . . . 76
4.2.8 Identity and Signature of the Empowered Person . . . 76

**A Safety Assessment** **77**
A.1 CE certification of the Robot Installation . . . 78
A.1.1 Safety Requirements . . . 78
A.1.2 Scoring the Risk . . . 78
A.1.3 Risk Assessment . . . 79
A.1.4 Example . . . 79

# Chapter 1

# Getting started

## 1.1 Introduction

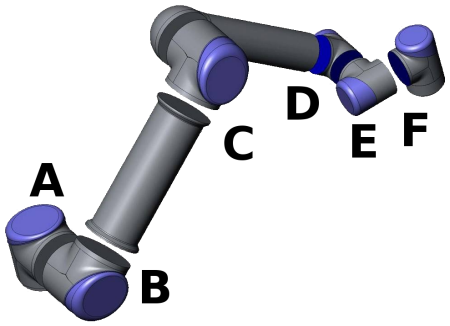
Congratulations on the purchase of your new Universal Robot, UR-6-85-5-A.

The robot is a machine that can be programmed to move a tool, and communicate with other machines using electrical signals. Using our patented programming interface, PolyScope, it is easy to program the robot to move the tool along a desired trajectory. PolyScope is described in section 3.1.

The reader of this manual is expected to be technically minded, to be familiar with the basic general concepts of programming, be able to connect a wire to a screw terminal, and be able to drill holes in a metal plate. No special knowledge about robots in general or Universal Robots in particular is required.

The rest of this chapter is an appetizer for getting started with the robot.

### 1.1.1 The Robot




The robot itself is an arm composed of extruded aluminum tubes and joints. The joints are named A:*Base*, B:*Shoulder*, C:*Elbow* and D,E,F:*Wrist 1,2,3*. The Base is where the robot is mounted, and at the other end (*Wrist 3*) the tool of the robot is attached. By coordinating the motion of each of the joints, the robot can move its tool around freely, with the exception of the area directly above and directly below the robot, and of course limited by the reach of the robot ($850mm$ from the center of the base).

### 1.1.2 Programs

A program is a list of commands telling the robot what to do. The user interface *PolyScope*, described later in this manual, allows people with only little programming experience to program the robot. For most tasks, programming is done entirely using the touch panel without typing in any cryptic commands.

Since tool motion is such an important part of a robot program, a way of teaching the robot how to move is essential. In PolyScope, the motions of the tool are given using a series of *waypoints*. Each waypoint is a point in the robot's workspace.

#### Waypoints

A waypoint is a point in the workspace of the robot. A waypoint can be given by moving the robot to a certain position, or can be calculated by software. The robot performs a task by moving through a sequence of waypoints. Various options regarding how the robot moves between the waypoints can be given in the program.

**Defining Waypoints, Moving the Robot.** The easiest way to define a waypoint is to move the robot to the desired position. This can be done in two ways: 1) By simply pulling the robot, while pressing the 'Back-drive' button on the touch screen (see 3.3.1). 2) By using the touch screen to drive the tool linearly or to drive each joint individually.

**Blends.** Per default the robot stops at each waypoint. By giving the robot freedom to decide how to move near the waypoint, it is possible to drive through the desired path faster without stopping. This freedom is given by setting a *blend radius* for the waypoint, which means that once the robot comes within a certain distance of the waypoint, the robot can decide to deviate from the path. A blend radius of 5-10 cm usually gives good results.

**Features**

Besides moving through waypoints, the program can send I/O signals to other machines at certain points in the robot's path, and perform commands like `if..then` and `loop`, based on variables and I/O signals.

### 1.1.3 Safety Evaluation

The robot is a machine and as such a safety evaluation is required for each installation of the robot. Appendix A.1 describes how to perform a safety evaluation.

## 1.2 Turning On and Off

How to turn the different parts of the robot system on and off is described in the following subsections.

### 1.2.1 Turning on the Controller Box

The controller box is turned on by pressing the 'On' button, at the front side of the controller box. When the controller box is turned on, a lot of text will appear on the screen. After about 30 seconds, the Universal Robot's Logo will appear, with the text 'Loading'. After around 70 seconds, a few buttons appear on the screen and a popup will force the user to go to the initialization screen.

### 1.2.2 Turning on the Robot

The robot can be turned on if the controller box is turned on, and if all emergency stop buttons are not activated. Turning the robot on is done at the initialization screen, by touching the 'ON' button at the screen. When it is turned on, a noise can be heard as the brakes unlock. After the robot has been turned on, it needs to be initialized before it can begin to perform work.

### 1.2.3 Initializing the Robot

After the robot is powered up, each of the robot's joints needs to find its exact position, by moving to a home position. Each large joint has around 20 home positions, evenly distributed over one joint revolution. The small joints have around 10. The Initialization screen, shown in figure 1.1, gives access to manual and semi-automatic driving of the robot's joints, to move them to a home position. The robot cannot automatically avoid collision with itself or the surrounds during this process. Therefor, caution should be exercised.

The *Auto* button near the top of the screen drives all joints until they are ready. When released and pressed again, all joints change drive direction. The *Manual* buttons permit manual driving of each joint.

A more detailed description of the initialization screen is found in section 3.1.2.

### 1.2.4 Shutting Down the Robot

The power to the robot can be turned off by touching the 'OFF' button at the initialization screen. Most users do not need to use this feature since the robot is

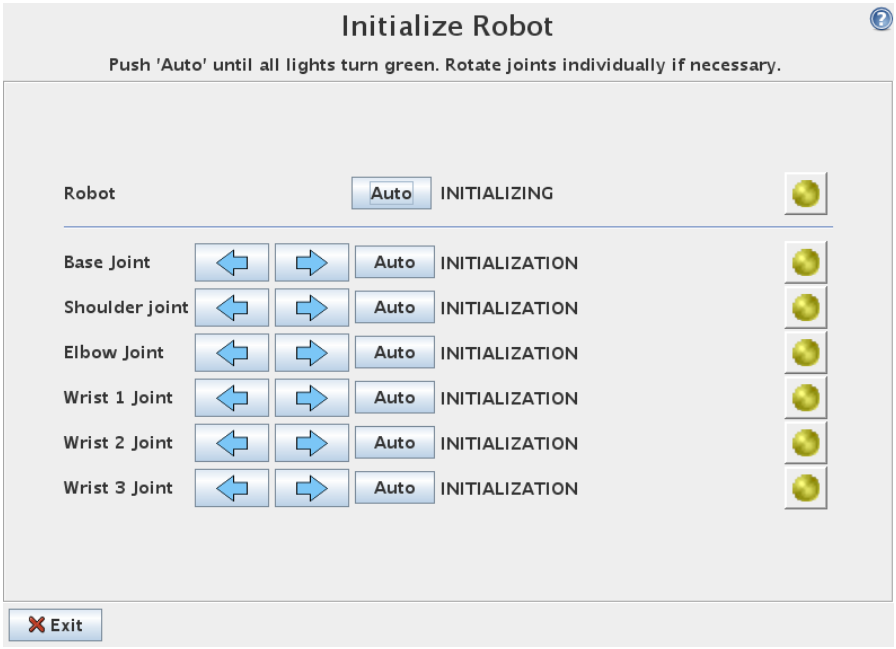



Figure 1.1: The initialization screen

automatically turned off when the controller box is shutting down. A third way is of course to push an emergency stop button.

### 1.2.5 Shutting Down the Controller Box

The proper way of shutting down the controller box is to use the on-screen menu system. Go to the 'File' menu at the top-left corner and choose 'Exit'. Then you see the 'Welcome' screen which has a 'Shut Down' button.

Shutting down by pulling the wall socket may cause corruption of the robot's file system, which may result in a robot malfunction. However, if the system locks up you can force a shutdown by pushing and holding the 'On' button at the front side of the controller box for five seconds.

## 1.3 Quick start, Step by Step

To quickly set up the robot, perform the following steps:

1. Unpack the robot and the controller box.
2. Mount the robot on a sturdy surface.
3. Place the controller box on its foot.
4. Plug the robot cable into the connector at the bottom of the controller box.
5. Plug in the mains connector of the controller box.
6. Press the Emergency Stop button on the front side of the controller box.
7. Press the power button next to the Emergency Stop button at the controller box.

8. Wait a minute while the system is starting up, displaying text on the touch screen.
9. When the system is ready, a popup will be shown on the touch screen, stating that the emergency stop button is pressed.
10. Touch the `To Initialization Screen` button at the popup.
11. Unlock the emergency stop buttons. The robot state then changes from 'Emergency Stopped' to 'Robot Power Off'.
12. Touch the `On` button on the touch screen. The robot now makes a noise and moves a little while unlocking the breaks.
13. Touch the blue arrows and move the joints around until every "light" at the right side of the screen turns green. Be careful not to drive the robot into itself or anything else.
14. All joints are now `OK`. Touch the exit button, bringing you the `Welcome` screen.
15. Touch the `PROGRAM Robot` button and select `Empty Program`.
16. Touch the `Next` button (bottom right) so that the `<empty>` line is selected in the tree structure on the left side of the screen.
17. Go to the `Structure` tab.
18. Touch the `Move` button.
19. Go to the `Command` tab.
20. Press the `Next` button, to go to the `Waypoint` settings.
21. Press the `Set this waypoint` button next to the `"?"` picture.
22. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Back-drive` button while pulling the robot arm.
23. Press `OK`.
24. Press `Add waypoint before`.
25. Press the `Set this waypoint` button next to the `"?"` picture.
26. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Back-drive` button while pulling the robot arm.
27. Press `OK`.
28. Your program is ready. The robot will move between the two points when you press the 'Play' symbol. Stand clear, hold on to the emergency stop button and press 'Play'.
29. Congratulations! You have now produced your first robot program that moves the robot between the two given positions. Remember that you have to carry out a risk assessment and improve the overall safety condition before you really make the robot do some work.

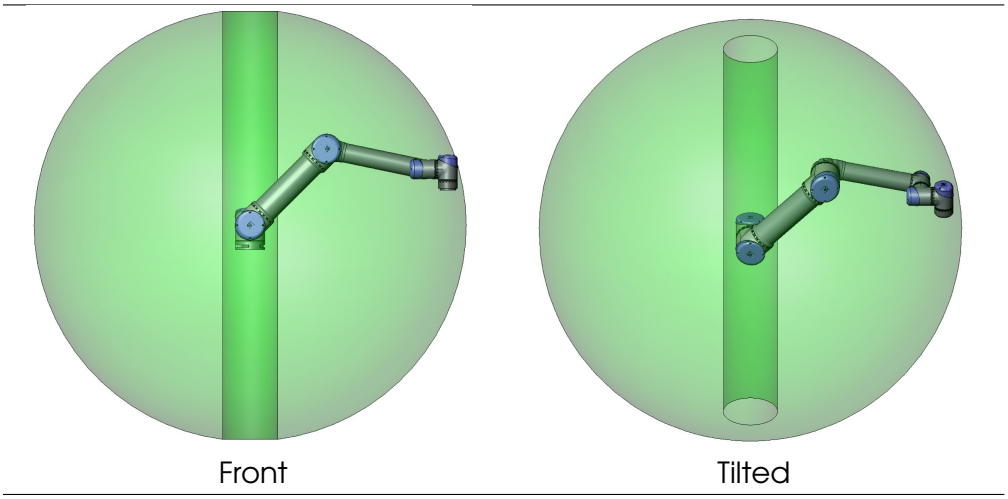



Figure 1.2: The workspace of the robot. The robot can work in an appoximate sphere (Ø170cm) around the base, except for a cylindrical volume directly above and directly below the robot base.

## 1.4 Mounting Instructions

The robot consists essentially of six robot joints and two aluminum tubes, connecting the robot's *base* with the robot's *tool*. The robot is built so that the tool can be translated and rotated within the robot's workspace. The next subsections describes the basic things to know when mounting the different parts of the robot system.

### 1.4.1 The Workspace of the Robot

The workspace of the UR-6-85-5-A robot extends to 850 mm from the base joint. The workspace of the robot is shown in figure 1.2. It is important to consider the cylindrical volume directly above and directly below the robot base when a mounting place for the robot is chosen. Moving the tool close to the cylindrical volume should be avoided if possible, because it causes the robot joints to move fast even though the tool is moving slowly.

### 1.4.2 Mounting the Robot

The robot is mounted using 4 M8 bolts, using the four $8.5mm$ holes on the robot's base. If very accurate repositioning of the robot is desired, two Ø8 holes are provided for use with a pin. Figure 1.3 shows where to drill holes and mount the screws.

### 1.4.3 Mounting the Tool

The robot tool flange has four holes for attaching a tool to the robot. A drawing of the tool flange is shown in figure 1.4.

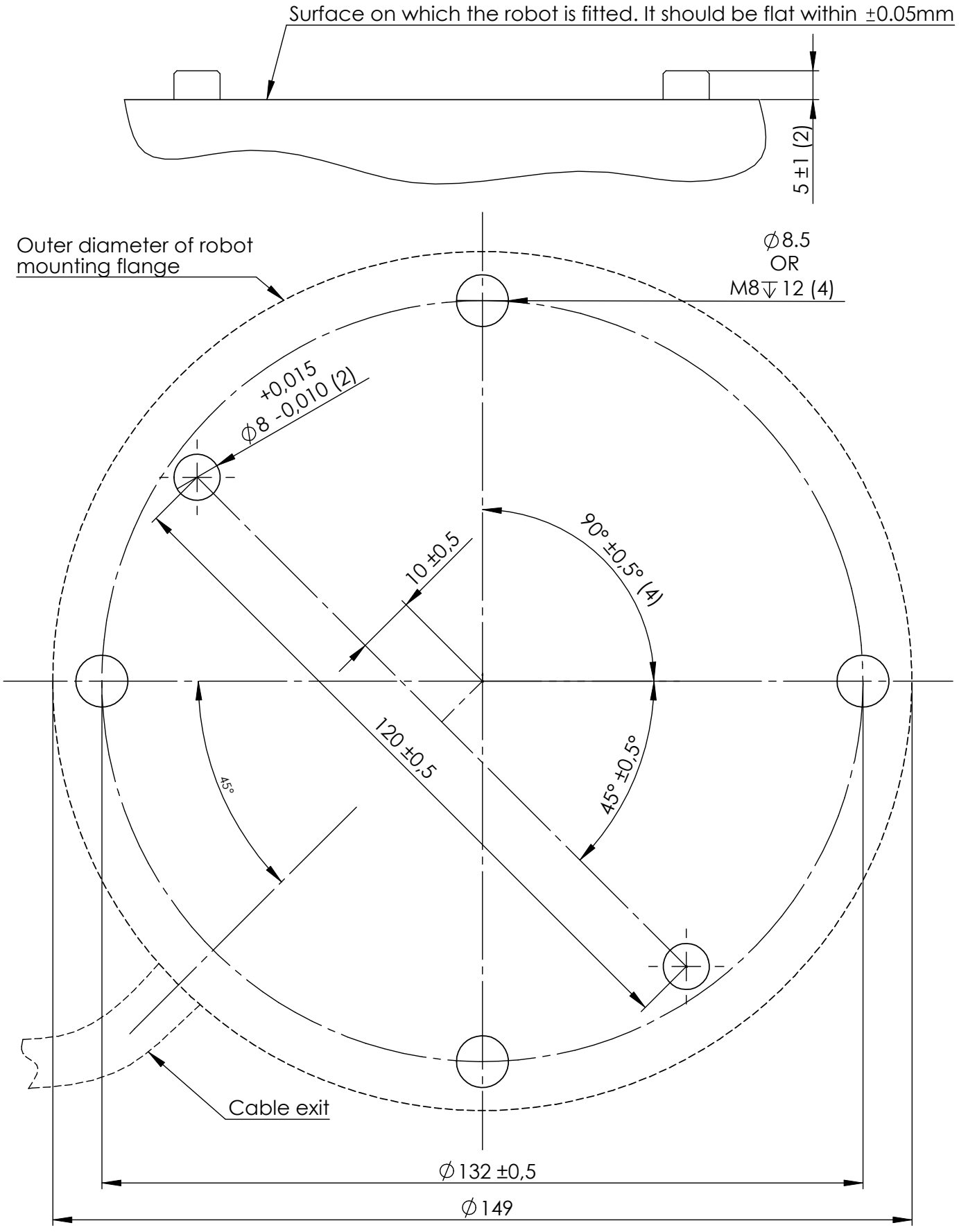



Figure 1.3: Holes for mounting the robot, scale 1:1. Use 4 M8 bolts. All measurements are in $mm$.

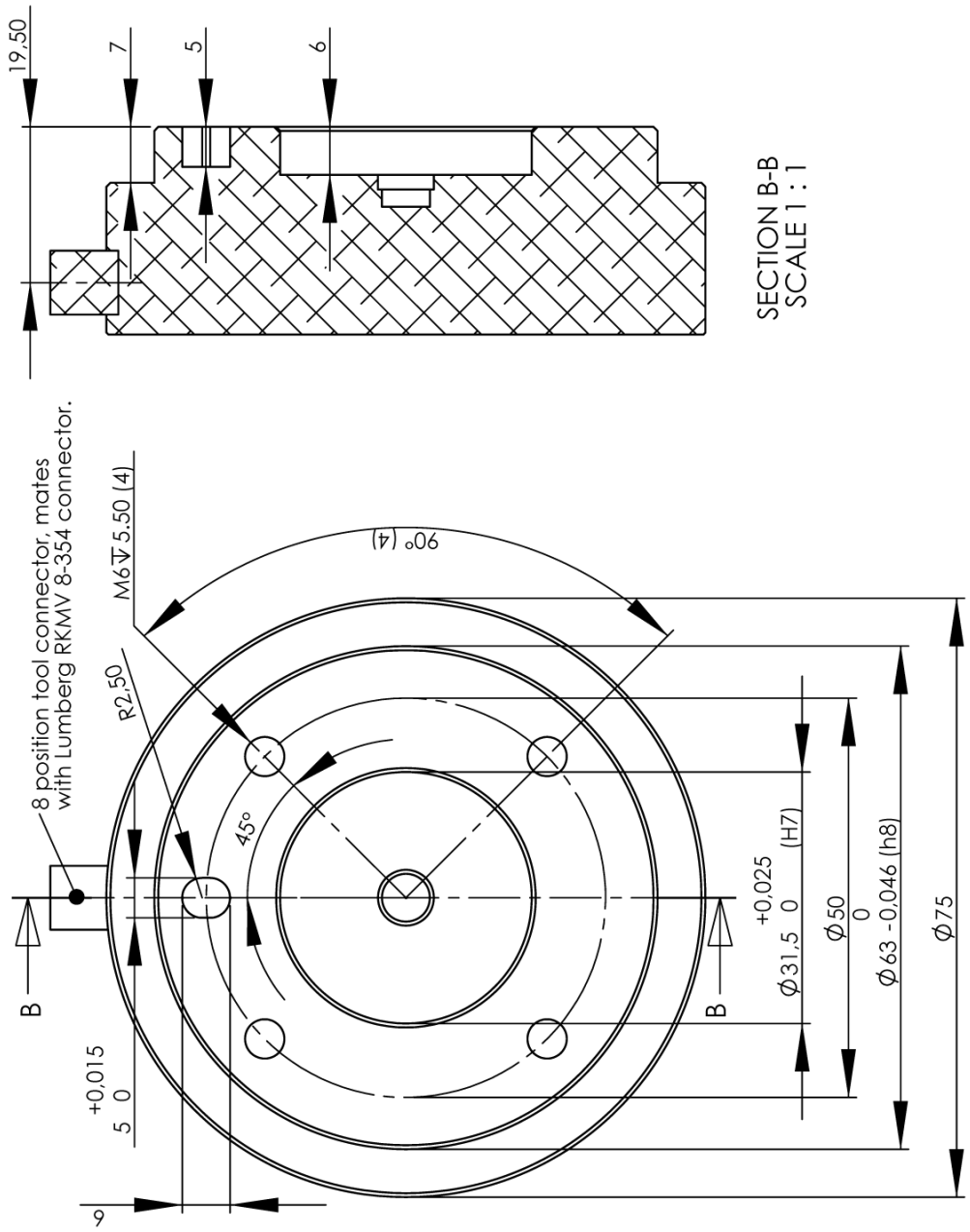



Figure 1.4: The tool output flange, EN ISO 9409-1-A50. This is where the tool is mounted at the tip of the robot. All measures are in $mm$.

| | |
|---|---|
| Input 100-120VAC<br>Input 200-240VAC | Min. 16A current rating<br>Min. 8A current rating |
| Frequency | 50-60Hz |
| Stand-by Power<br>Typical 'On' Power | 5W<br>200W |

Table 1.1: Specifications for mains connection

### 1.4.4 Mounting the Controller Box

The controller box can be mounted using the two holes on the back of the controller box, or it can be placed on the ground.

### 1.4.5 Mounting the Touch Panel

The touch sensitive screen can be hung on a wall or on the controller box. Extra fittings can be bought.

### 1.4.6 Connecting the Robot Cable

The cable from the robot must be plugged in to the connector at the button of the controller box. Ensure that the connector is properly locked. Connecting and disconnecting the robot cable may only be done when the robot power is turned off, which is easily ensured by pushing the emergency stop button on the front side of the controller box.

### 1.4.7 Connecting the Mains Cable

The mains cable from the controller box has a standard IEC plug in the end. Connect a country specific mains plug or cable to the IEC plug. Remember to use a cable with specifications as shown with the mains specifications in table 1.1.

The controller box should be connected to earth by the mains cable. If other earth connections are needed for external equipment, please use the M8 screw at the bottom right corner of the controller box, as shown below.

230V
AC

# Chapter 2

# Electrical Interface

## 2.1 Introduction

There are electrical inputs/outputs (I/Os) inside the controller box and at the robot tool flange. Some of the I/Os inside the controller box are dedicated to the robot emergency stop functionality, and some I/Os allows the robot to communicate with other machines and equipment. The I/O at the robot tool flange can be used to control grippers and sensors placed on the tool. Both the controller and the tool I/O can be tested at the I/O tab in the graphical user interface, as explained in section 3.3.2. The next three sections explain how to use the electrical I/O.

Note that according to the IEC 61000 standard cables going from the controller box to other machinery and factory equipment may not be longer than 30m, unless extended test requirements are performed.

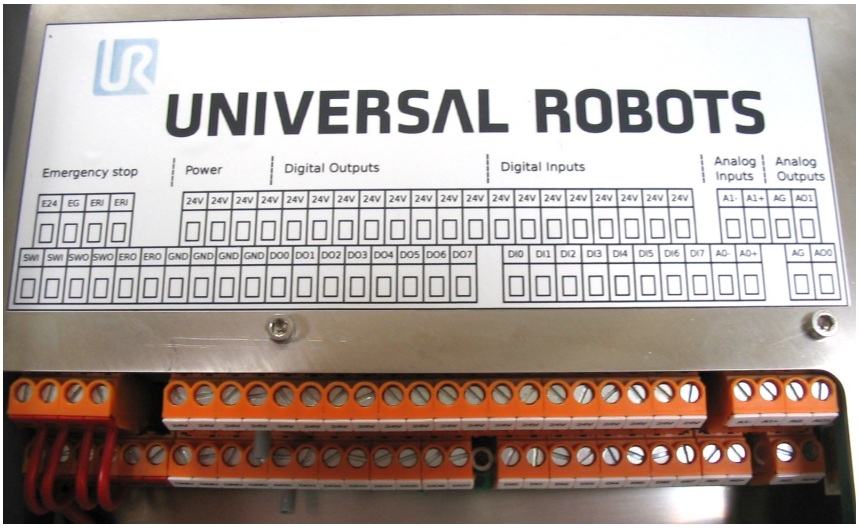
Note that every minus connection (0V) is referred to as GND, and is connected to the shield of the robot and the controller box. However, all mentioned GND connections are only for powering and signaling. For a real ground connection there is an M 10 sized screw connection at the down right corner of the controller box.

Note that data in this chapter is only valid when the ambient temperature of the controller box and the robot is within its specified working range, and that all voltage and current data is implicitly DC.

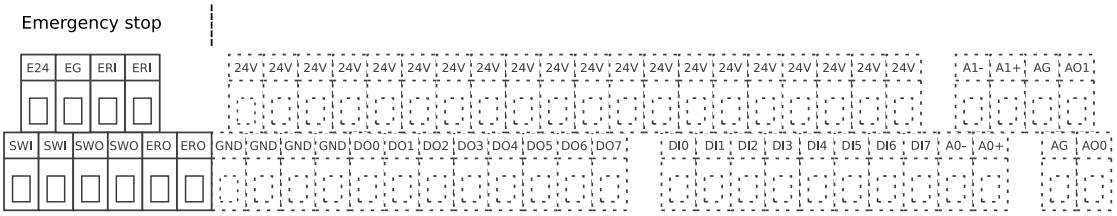
| | |
|---|---|
| E24 | 24V Emergency stop power supply |
| EG | 0V Emergency stop GND connection |
| SWI | Emergency stop button switch input |
| SWO | Emergency stop button switch output |
| ERI | Emergency relay input |
| ERO | Emergency relay output |

Table 2.1: Abbreviations for the emergency stop interface

## 2.2 The Emergency Stop Interface




Inside the controller box there is a panel of screw terminals as shown above. It is only the leftmost part which is used for the emergency stop functions; the other terminals are normal I/O as shown below.


Emergency stop
E24 EG ERI ERI
SWI SWI SWO SWO ERO ERO
GND GND GND GND DO0 DO1 DO2 DO3 DO4 DO5 DO6 DO7
24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V 24V
DI0 DI1 DI2 DI3 DI4 DI5 DI6 DI7 A0- A0+
A1- A1+ AG AO1
AG AO0

The abbreviations are explained in table 2.1.

Note that connecting and configuring the emergency interface relies on the complete understanding of the emergency circuitry, and the owner of the machinery takes full responsibility for connecting it correctly and to the right safety level.

Note the number of safety components that should be used and how they must work rely on the risk assessment, which is explained in section A.1.

Note that it is important to make regular checks of the emergency stop functionality to ensure that all emergency stop devices are functioning correctly.

The emergency stop interface is different from the normal I/O, because it has to comply with a certain safety level (EN 954 Category 3). To understand the emergency stop functionality, a simplified version of the internal schematics of the robot emergency stop circuitry is shown in figure 2.1. It is important to notice that any short circuit or lost connection will lead to an emergency stop,

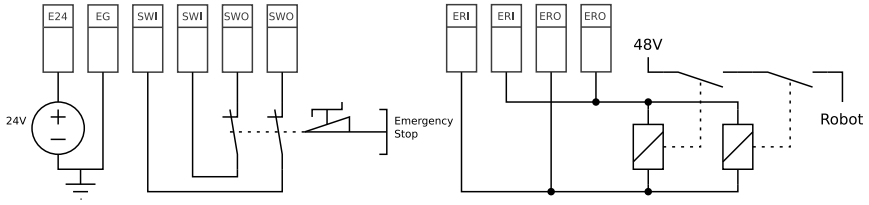



Figure 2.1: Simplified schematics of the internal robot emergency stop circuitry.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Voltage available at connection E24 | TBD | 24 | TBD | V |
| Current available at connection E24 | - | - | 800* | mA |
| Short-circuit current protection | - | 850 | - | mA |
| Capacitive load at connection E24 | - | - | TBD | uF |
| Inductive load at connection E24 | - | - | TBD | uH |
| Emergency relay ON voltage | 18 | 24 | 26 | V |
| Emergency relay OFF voltage | - | 0 | 1.5 | V |
| Emergency relay quiescent current | - | 110 | TBD | mA |
| Current through internal switch | - | - | 1.0 | A |

Table 2.2: Emergency stop interface data. TBD = To Be Determined.

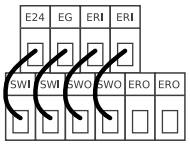
as long as only one error appears at a time. Failure and abnormal behavior of relays and power supplies results in an error message in the robot log and prevents the robot from powering up.

It is generally important that the connections to the emergency stop interface are separated from the normal I/O interface, and that it is never connected to a PLC which is not a safety PLC with the right safety level. If this rule is not followed, it is not possible to get a high safety level, because one failure in normal I/O can prevent an emergency stop signal from resulting in an emergency stop. Other rules that restrict the use of the emergency stop interface are shown in table 2.2.

Note that connection E24 is sourced by the same internal 24V regulator as the normal I/O, and that the maximum of 800mA is for both power sources together.

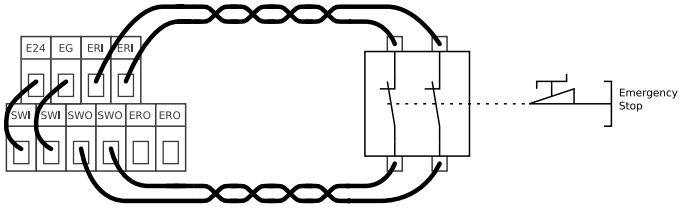
The internal control system will power off the robot if the current exceeds its limit. This will also generate an error message in the robot log. The next subsections show some simple examples of how the emergency stop interface can be connected to other safety equipment and other safety circuits.

### 2.2.1 The Simplest Emergency Stop Configuration



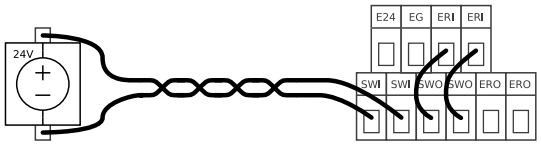

The simplest configuration is to use the internal emergency stop button as the only component to generate an emergency stop. This is done with the configuration shown above. This configuration is the default when the robot leaves the factory, and thereby the robot is ready to operate. However, the emergency configuration should be changed if required by the risk assessment.

### 2.2.2 Connecting an External Emergency Stop Button




In almost every robot application it is required to connect one or more external emergency stop buttons. Doing so is simple and easy. An example of how to connect one extra button is shown above. Remember that only approved emergency stop buttons with double switches are good enough. It is also possible to connect light curtains and door switches etc., as long as the equipment is approved for emergency stop with the right safety level.

### 2.2.3 Using an External Emergency Stop Power Supply

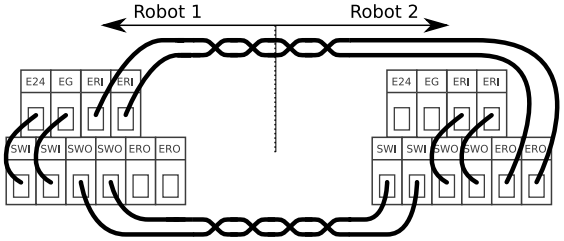



If the robot is part of a bigger system, it is sometimes preferred or required to use an external source of 24V for the emergency stop circuitry. How to connect an external source is shown above.

### 2.2.4 Connecting to Other Machinery

When the robot is used together with other electro-mechanical machinery, it is often required to set up a common emergency circuit. This ensures that if a dangerous situation arises, the operator does not need to think about which buttons to use. It is also often preferable for every part of a sub-function in a product line to be synchronized, since a stop in only one part of the product line can lead to a dangerous situation.

A UR robot uses simple 24V signals for emergency signaling as does most industrial machinery. It is therefore possible to connect the controller box to most industrial machinery, without using any special and expensive equipment, such as safety approved relays and PLCs. The principle is to choose a common 24V voltage source, and connect all emergency stop button in series, and then all the relays of the machinery. An example with two UR robots is shown below.




Remember to check that all emergency stop buttons are rated for the total current consumption of all the connected emergency stop relays.

## 2.3 The Pause Interface

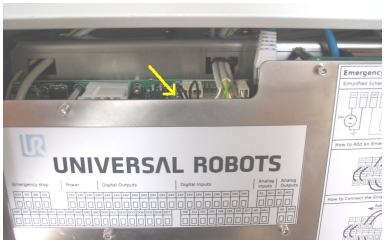
Note: The pause interface can at most be used as a category 1 safeguard interface, depending on the external wiring.

Using the pause interface, the robot program can pause due to an external event. The external event can be caused by a light braker circuit, a pressure sensitive floor mat or a similar device that can give a signal when a person is near the robot. When paused, the program can be resumed without loss of program state. To resume the program, click "Continue" on the Popup on the screen.

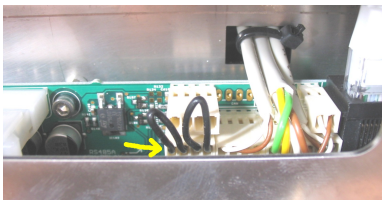
### 2.3.1 Connecting to the Pause Interface

Install the pause interface as shown.

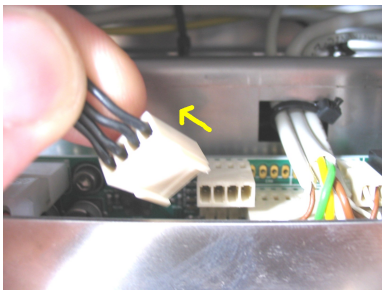
You need a Pause connector.



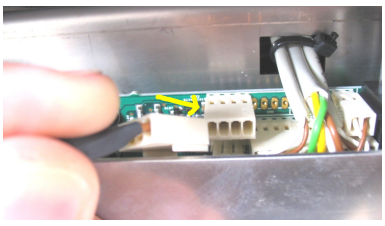

Open the controller box and look near the top.



Locate the Pause placeholder plug.

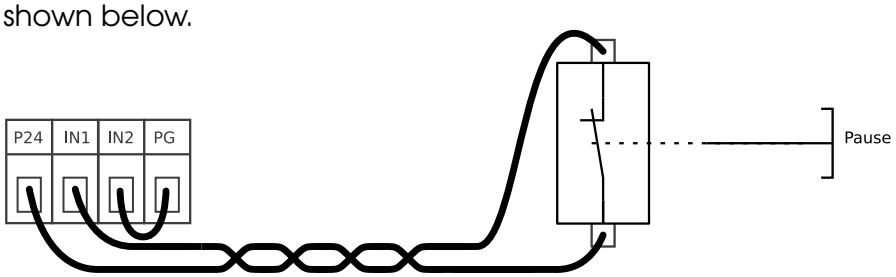


Remove the placeholder.

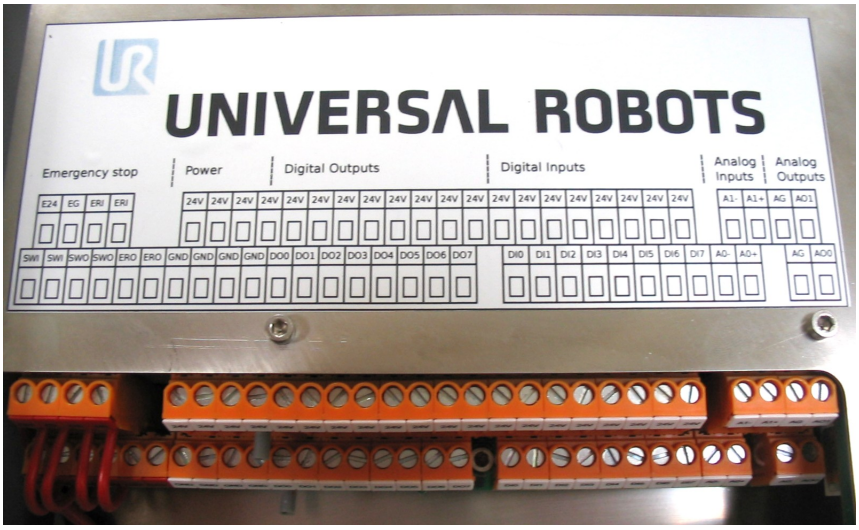


Plug in the Pause connector.

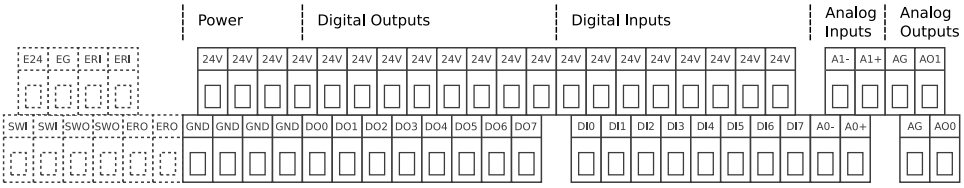
When the pause connector is in place, a pause device can be wired as shown below.



## 2.4 Controller I/O




Inside the controller box there is a panel of screw terminals with various I/O parts, as shown above. The leftmost part of this panel is used for the emergency stop functionality, as shown below.


Power
Digital Outputs
Digital Inputs
Analog Inputs
Analog Outputs
E24 EG ERI ERI
24V
SWI SWI SWO SWO ERO ERO GND GND GND GND DO0 DO1 DO2 DO3 DO4 DO5 DO6 DO7
DI0 DI1 DI2 DI3 DI4 DI5 DI6 DI7
A1- A1+ AG AO1
A0- A0+
AG AO0

Note that any change in the emergency stop circuitry can lead to a dangerous robot condition, even though the robot emergency stop functionality seems to be present. Never combine the emergency stop circuit with the normal I/O.

The abbreviations of the I/O panel are explained in table 2.3.

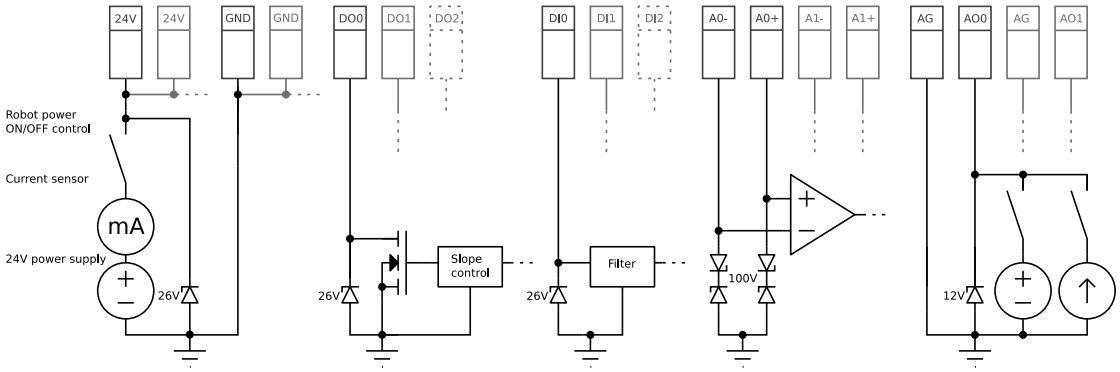
| | |
|---|---|
| 24V<br>GND | 24V power supply<br>0V GND connection |
| DOx<br>DIx | Digital output number x<br>Digital input number x |
| AOx<br>AG | Analog output number x plus<br>Analog output GND |
| Ax+<br>Ax- | Analog input number x plus<br>Analog input number x minus |

Table 2.3: Abbreviations for the I/O interface inside the controller box.

To get a good understanding of the I/O interface, a simplified version of the internal circuitry is shown below.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Voltage available at connection 24V | TBD | 24 | TBD | V |
| Current available at connection 24V | - | - | 800* | mA |
| Short-circuit current protection | - | 850 | - | mA |
| Capacitive load at connection 24V | - | - | TBD | uF |
| Inductive load at connection 24V | - | - | TBD | uH |

Table 2.4: Normal I/O interface data. TBD = To Be Determined.




The left part shows the general purpose 24V power supply, which the user can use for basic controlling and powering. Note that the 24V is only turned on when the robot is turned on. This also means that if an operator pushes the emergency stop button, then the power disappears. Just remember that the 24V may not source or control any functions which can lead to dangerous situations according to the risk assessment.

The general data on the 24V power supply is shown in table 2.4.

Note that connection E24 is sourced by the same internal 24V regulator as the normal I/O, and that the maximum of 800mA is for both power sources together.

The internal control system will power off the robot if the current exceeds its limit. This will also generate an error message in the robot log. The next subsections show some simple examples of how to use the different I/O functionalities.

### 2.4.1 Digital Outputs

The digital outputs are implemented so that they can only sink to GND (0V) and not source current. When a digital output is activated, the corresponding connection is driven to GND, and when it is deactivated, the corresponding connection is open (open-collector/open-drain). The advantage of this implementation is that it is possible to use any external power supply instead of the internal 24V power supply, as long as its voltage is not higher than the specified limit.

The digital outputs are limited by the data specified in table 2.5.

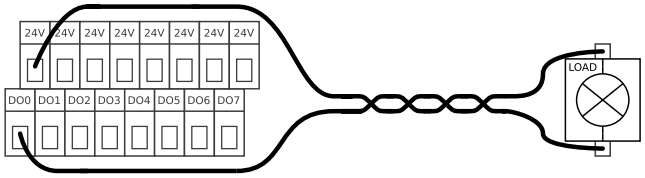
Note that the digital outputs are not current limited and overriding the specified data can cause permanent damage.

To illustrate clearly how to use the digital output ports, some simple examples are shown.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Voltage when open | -0.5 | - | 26 | V |
| Voltage when sinking 1A | - | 0.05 | 0.20 | V |
| Current when sinking | 0 | - | 2 | A |
| Current through one screw terminal | - | - | 10 | A |
| Switch time for DO0 to DO5 | - | 500 | - | uS |
| Switch time for DO6 to DO7 | - | 10 | - | uS |
| Capacitive load | - | - | TBD | uF |
| Inductive load | - | - | TBD | uH |

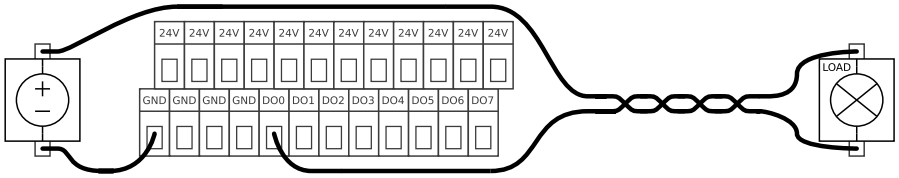
Table 2.5: Data specification of digital outputs. TBD = To Be Determined.

#### Load Controlled by Digital Output




This example illustrates how to turn on a load, when using the internal 24V power supply. Remember that there are 24V between the 24V connection and the shield/ground, even when the load is turned off.

#### Load Controlled by Digital Output, External Power




If the available current from the internal power supply is not enough, or if the load needs another voltage such as 12V, simply use an external power supply, as shown above.

Another basic way to use digital outputs is to communicate with other industrial equipment such as PLCs or another UR robot. An example of this is shown in the next section, which describes the digital inputs.

### 2.4.2 Digital Inputs

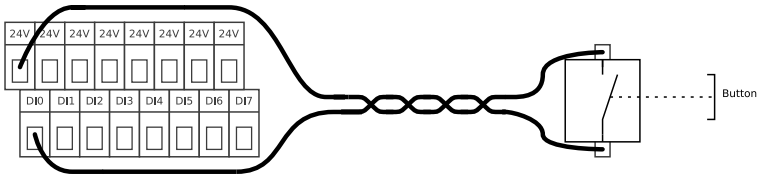
The digital inputs are implemented with weak pull-down resistors. This means that a floating input will always read low. The voltages at which the inputs are guaranteed to read low or high are shown with the other data in table 2.6.

To make it clear how easy it is to use digital inputs, some simple examples are shown.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Input voltage | -0.5 | - | 26 | V |
| Logical low voltage | - | - | 2.0 | V |
| Logical high voltage | 5.5 | - | - | V |
| Input resistance | - | 47k | - | ohm |

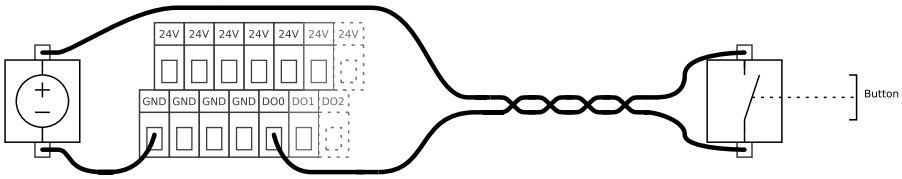
Table 2.6: Data specification of digital inputs.

### Digital Input, Simple Button




The above example shows how to connect a simple button or switch. A bad quality switch might trigger the input twice due to a long mechanical stabilizing time of the two conducting surfaces. However, in most programs it will not cause problems.

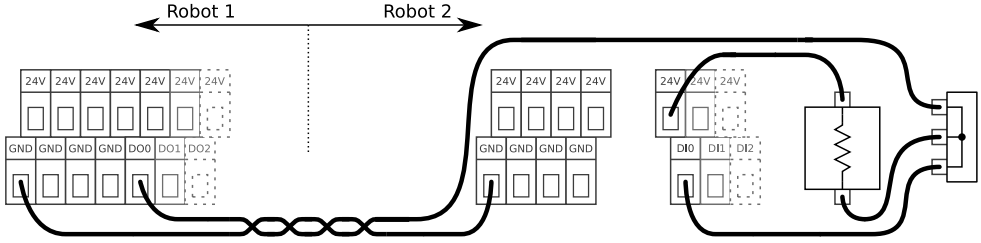
### Digital Input, Simple Button




The above illustration shows how to connect a button using an external power source. Remember that table 2.6 specifies the valid supply voltage for this case.

### Signal Communication with other Machinery or PLCs

If communication with other machinery or PLCs is needed, and the signal driver is both sinking and sourcing, communication is done by direct wiring. Since the digital outputs of a UR robot are only sinking, a pull-up resistor is needed. An example where two UR robots are communicating with each other is illustrated below.




The UR robot on the left side is communicating with the robot on the right side. A typical value for the resistor shown is 10kohm. The three-terminal box is just a terminal strip.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Valid output voltage in current mode | 0 | - | 10 | V |
| Valid output current in voltage mode | -20 | - | 20 | mA |
| Short-circuit current in voltage mode | - | 40 | - | mA |
| Output resistance in voltage mode | - | 43 | TBD | ohm |
| Offset error @ 4mA, load = 500ohm | - | - | TBD | mA |
| Total error @ 20mA, load = 500ohm | - | - | TBD | mA |
| Offset error @ 0V, load = 1Mohm | - | 0.5 | TBD | mV |
| Total error @ 5V, load = 1Mohm | - | 50 | TBD | mV |

Table 2.7: Data specification of analog outputs. TBD = To Be Determined.

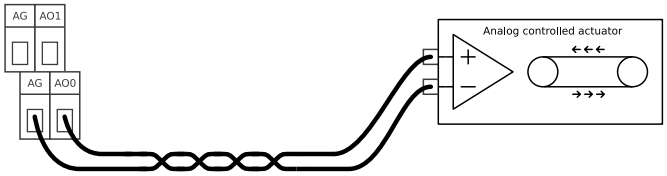
Note that if the robot on the left side of the illustration is turned off, the input signal of the right robot will be high, and this can lead to unexpected behavior. Combining the emergency stop circuitry between the robots should be considered to avoid these situations.

### 2.4.3 Analog Outputs

The analog outputs can be set for both current mode and voltage mode, in the range of 4-20mA and 0-5V respectively. The analog outputs are limited by the data shown in table 2.7.

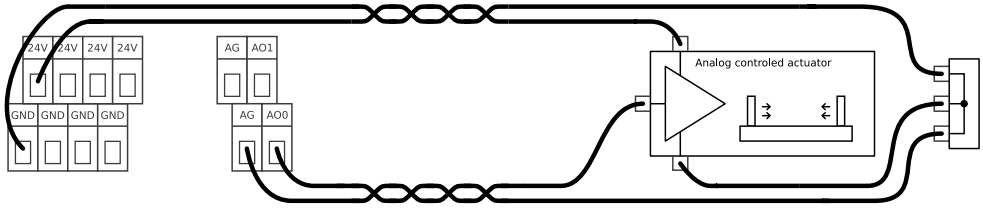
To illustrate clearly how easy it is to use analog outputs, some simple examples are shown.

#### Using the Analog Outputs




This is the normal and best way to use analog outputs. The illustration shows a setup where the robot controller controls an actuator like a conveyor belt. The best result is accomplished when using current mode, because it is more immune to disturbing signals.

#### Using the Analog Outputs, Non-Differential Signal




If the controlled equipment does not take a differential input, an alternative solution can be made as shown above. This solution is not very good in terms of noise, and can easily pick up disturbing signals from other machinery. Care must be taken when the wiring is done, and it must be kept in mind that disturbing signals induced into analog outputs may also be present on other analog I/O.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Common mode input voltage | -90 | - | 90 | V |
| Differential mode input voltage* | -120 | - | 120 | V |
| Differential input resistance | - | 220 | - | kohm |
| Common mode input resistance | - | 55 | - | kohm |
| Common mode rejection ratio | 75 | - | - | dB |
| Offset error @ Range 0 - 5 | - | TBD | TBD | mV |
| Offset error @ Range 0 - 10 | - | TBD | TBD | mV |
| Offset error @ Range -5 - 5 | - | TBD | TBD | mV |
| Offset error @ Range -10 - 10 | - | TBD | TBD | mV |
| Total error @ Range 0 - 5 | - | TBD | TBD | mV |
| Total error @ Range 0 - 10 | - | TBD | TBD | mV |
| Total error @ Range -5 - 5 | - | TBD | TBD | mV |
| Total error @ Range -10 - 10 | - | TBD | TBD | mV |

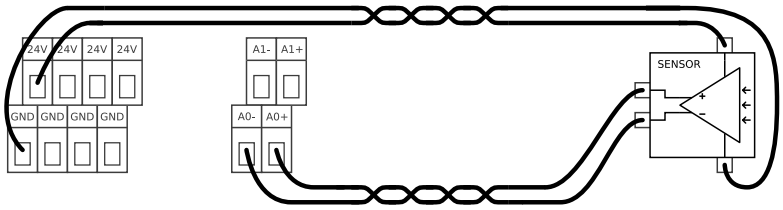
Table 2.8: Data specification of analog inputs. TBD = To Be Determined.

### 2.4.4 Analog Inputs

The analog inputs can be set to four different voltage ranges, which are implemented in different ways, and therefore can have different offset and gain errors. The technical data defining limitations on the analog inputs are shown in table 2.8.

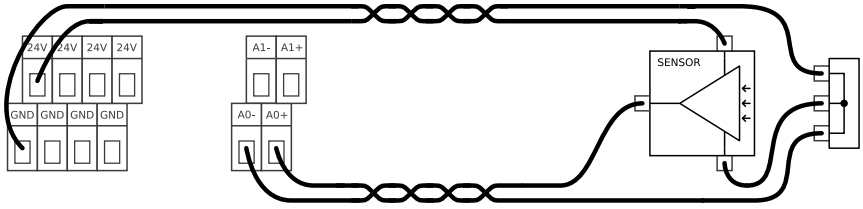
The specified differential mode input voltage is only valid with a common mode voltage of 0V. To make it clear how easy it is to use analog outputs, some simple examples are shown.

#### Using Analog Inputs, Differential Voltage Input



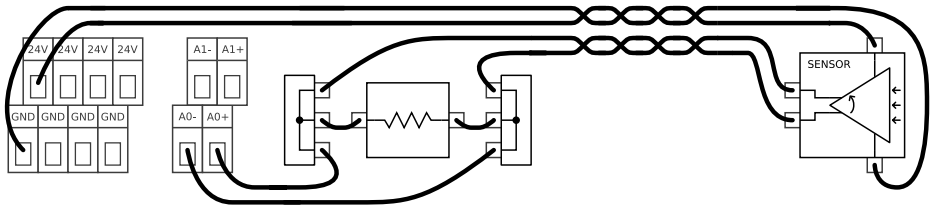

The simplest way to use analog inputs. The equipment shown, which could be a sensor, has a differential voltage output.

#### Using Analog Inputs, Non-differential Voltage Input




If it is not possible to achieve a differential signal from the equipment used, a solution could look something like the setup above. Unlike the non-differential analog output example in subsection 2.4.3, this solution would be almost as good as the differential solutions.

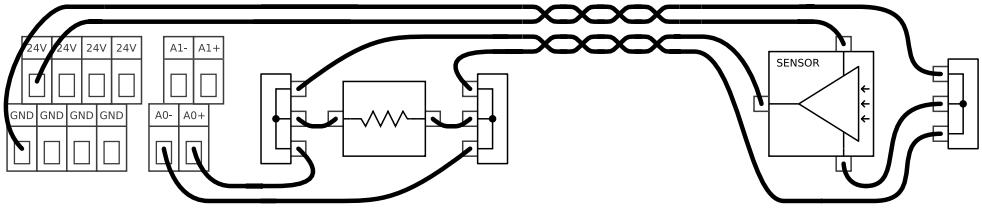
### Using Analog Inputs, Differential Current Input




When longer cables are used, or if it is a very noisy environment, current based signals are preferred. Also, some equipment comes only with a current output. To use current as inputs, an external resistor is needed as shown above. The value of the resistor would normally be around 200 ohms, and the best result is accomplished when the resistor is close to the screw terminals of the controller.

Note that the tolerance of the resistor and the ohmic change due to temperature must be added to the error specifications of the analog inputs.

### Using Analog Inputs, Non-differential Current Input




If the output of the equipment is a non-differential current signal, a resistor must be used as shown above. The resistor should be around 200 ohms and the relationship between the voltage at the controller input and the output of the sensor is given by:

$$\text{Voltage} = \text{Current} \times \text{Resistance}$$

Note that the tolerance of the resistor and the ohmic change due to temperature must be added to the error specifications of the analog inputs.

## 2.5 Tool I/O

At the tool end of the robot there is a small connector with eight connections.

| Colour | Signal |
|---|---|
| Red | 0V (GND) |
| Gray | 0V/12V/24V (POWER) |
| Blue | Digital output 8 (DO8) |
| Pink | Digital output 9 (DO9) |
| Yellow | Digital input 8 (DI8) |
| Green | Digital input 9 (DI9) |
| White | Analog input 2 (AI2) |
| Brown | Analog input 3 (AI3) |

Table 2.9: Relation between cable colours and functions.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Supply voltage in 24V mode | TBD | 24 | TBD | V |
| Supply voltage in 12V mode | TBD | 12 | TBD | V |
| Supply current in both modes | - | - | 600 | mA |
| Short-circuit current protection | - | 650 | - | mA |
| Capacitive load | - | - | TBD | uF |
| Inductive load | - | - | TBD | uH |

Table 2.10: Data specification of tool power supply. TBD = To Be Determined.

This connector provides power and control signals for basic grippers and sensors, which may be present at on specific robot tool. The reason for having this connector is to save the wiring between the tool and the controller box. It is of course necessary to add wires if the I/O provided is insufficient. The connector is a standard Lumberg RSMEDG8, which mates with a cable named RKMV 8-354. Table 2.9 shows the different I/O and the corresponding cable colors.

Note that the tool flange is connected to GND (same as the red wire).

The available power supply can be set to either 0V, 12V or 24V at the I/O tab in the graphical user interface (see section 3.3.2). Take care when using 12V, since an error made by the programmer can cause a voltage change to 24V, which might damage the equipment and even cause a fire. The specifications on the power supply are shown in Table 2.10.

The internal control system will generate an error to the robot log if the current exceeds its limit. The different I/Os at the tool is described in the following three subsections.

### 2.5.1 Digital Outputs

The digital outputs are implemented so that they can only sink to GND (0V) and not source current. When a digital output is activated, the corresponding connection is driven to GND, and when it is deactivated, the corresponding connection is open (open-collector/open-drain). The primary difference between the digital outputs inside the controller box and those in the tool is the reduced current due to the small connector. Table 2.11 lists the specified data.

Note that the digital outputs in the tool are not current limited and overriding the specified data can cause permanent damage.

To illustrate clearly how easy it is to use digital outputs, a simple example is shown.

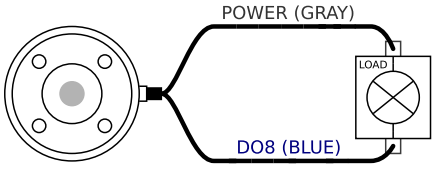
| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Voltage when open | -0.5 | - | 26 | V |
| Voltage when sinking 1A | - | 0.05 | 0.20 | V |
| Current when sinking | 0 | - | 1 | A |
| Current through GND | - | - | 1 | A |
| Switch time | - | 1000 | - | us |
| Capacitive load | - | - | TBD | uF |
| Inductive load | - | - | TBD | uH |

Table 2.11: Data specification of digital outputs. TBD = To Be Determined.

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Input voltage | -0.5 | - | 26 | V |
| Logical low voltage | - | - | 2.0 | V |
| Logical high voltage | 5.5 | - | - | V |
| Input resistance | - | 47k | - | ohm |

Table 2.12: Data specification of digital inputs.

**Using Digital Outputs**



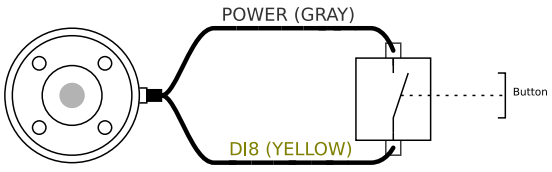

This example illustrates how to turn on a load, when using the internal 12V or 24V power supply. Remember that you have to define the output voltage at the I/O tab (see section 3.3.2). Keep in mind that there is voltage between the POWER connection and the shield/ground, even when the load is turned off.

### 2.5.2 Digital Inputs

The digital inputs are implemented with weak pull-down resistors. This means that a floating input will always read low. The digital inputs at the tool are implemented in the same way as the digital inputs inside the controller box. The voltages at which the inputs are guaranteed to read low or high are shown with the other data in table 2.12.

To illustrate clearly how easy it is to use digital outputs, a simple example is shown.

**Using Digital Inputs**




The above example shows how to connect a simple button or switch. A bad quality switch might trigger the input twice due to a long mechanical stabilizing

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Input voltage in voltage mode | -0.5 | - | 26 | V |
| Input voltage in current mode | -0.5 | - | 5.0 | V |
| Input current in current mode | -2.5 | - | 25 | mA |
| Input resistance @ range 0V to 5V | - | 29 | - | kohm |
| Input resistance @ range 0V to 10V | - | 15 | - | kohm |
| Input resistance @ range 4mA to 20mA | - | 200 | - | ohm |
| Offset error @ Range 0 - 5 | - | TBD | TBD | mV |
| Offset error @ Range 0 - 10 | - | TBD | TBD | mV |
| Offset error @ Range 4mA to 20mA | - | TBD | TBD | mA |
| Total error @ Range 0 - 5 | - | TBD | TBD | mV |
| Total error @ Range 0 - 10 | - | TBD | TBD | mV |
| Total error @ Range 4mA to 20mA | - | TBD | TBD | mA |

Table 2.13: Data specification of analog inputs. TBD = To Be Determined.

time of the two conducting surfaces. However, in most programs it will not cause problems.

### 2.5.3 Analog Inputs

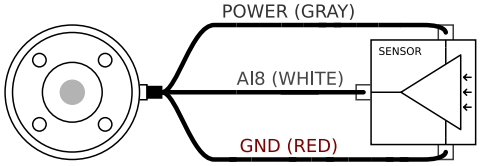
The analog inputs at the tool are very different from those inside the controller box. The first ting to notice is that they are non-differential, which is a drawback compared to the analog inputs at the controller I/O. The second thing to notice is that the tool analog inputs have current mode functionality, which is an advantage compared with the controller I/O. The analog inputs can be set to different input ranges, which are implemented in different ways, and therefore can have different offset and gain errors. The data specification of the analog inputs is shown in Table 2.11.

An important thing to realize is that any current change in the common GND connection can result a disturbing signal in the analog inputs, because there will be a voltage drop along the GND wires and inside connectors.

Note that a connection between the tool power supply and the analog inputs will permanently damage the I/O functionality, if the analog inputs are set in current mode.

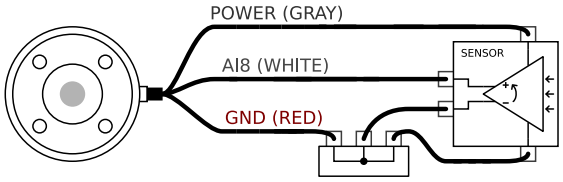
To make it clear how easy it is to use digital inputs, some simple examples are shown.

#### Using Analog Inputs, Non-differential




The simplest way to use analog inputs. The output of the sensor can be either current or voltage, as long as the input mode of that analog input is set to the same on the I/O tab (see section 3.3.2). Remember to check that a sensor with voltage output can drive the internal resistance of the tool, or the measurement might be invalid.

### Using Analog Inputs, Differential



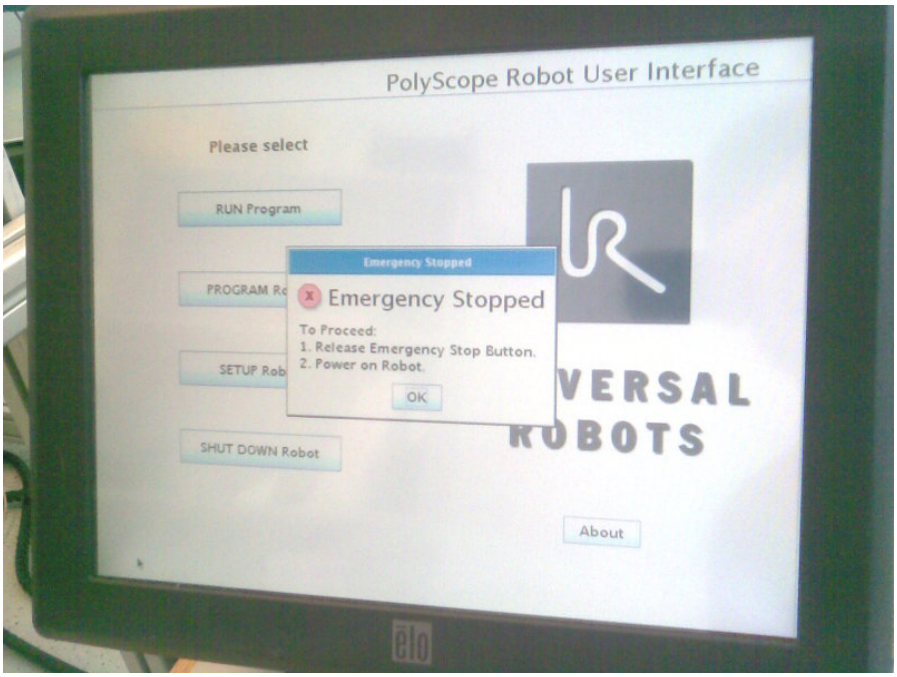

Using sensors with differential outputs is also straightforward. Simply connect the negative output part to GND (0V) with a terminal strip and it will work in the same way as a non-differential sensor.

# Chapter 3

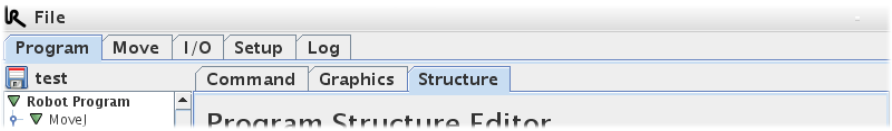
# PolyScope Software

## 3.1 Introduction

PolyScope is the graphical user interface (GUI) which lets you operate the robot, run existing robot programs or easily create new ones. PolyScope runs on the touch sensitive screen attached to the control box. To calibrate the touch screen, read section 3.5.5.




The picture above shows the Welcome Screen, with a popup saying that the robot is emergency stopped. The bluish areas of the screen are buttons that can be pressed by pressing a finger or the backside of a pen. PolyScope has a hierarchical structure of screens. In the programming environment, the screens are arranged in *tabs*, for easy access on the screens.



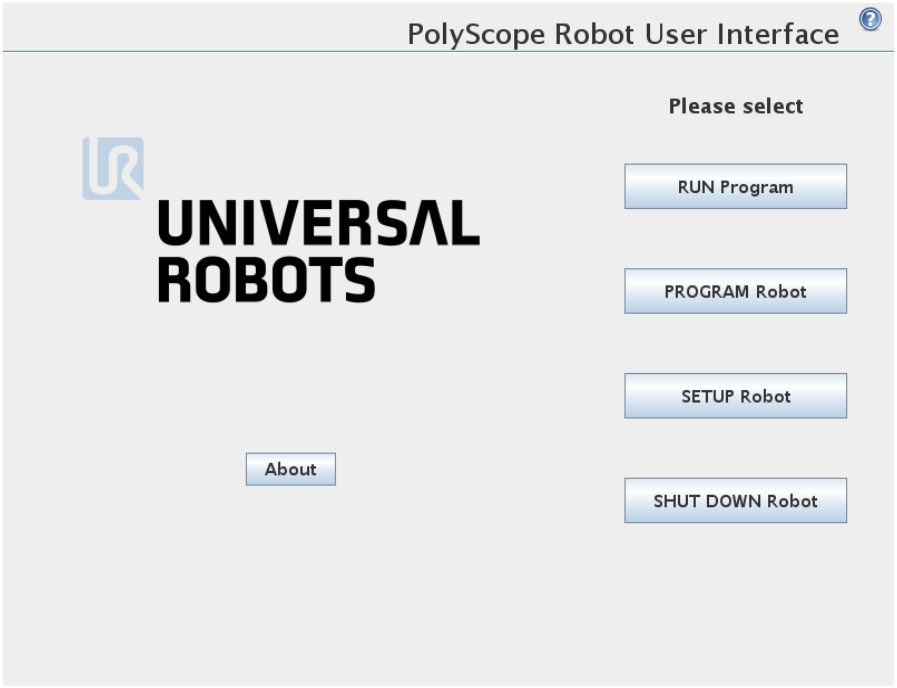

In this example, the `Program` tab is selected at the top level, and under that the `Structure` tab is selected. The `Program` tab holds information related to the currently loaded program. If the `Move` tab is selected, the screen changes to the 'Move' screen, from where the robot can be moved. Similarly, by selecting the `I/O` tab, the current state of the electrical I/O can be monitored and changed.

It is possible to connect a mouse and a keyboard to the controller box; however, this is not required. Whenever a text or number input is needed, an on-screen keypad or keyboard is provided.

The on-screen keypad, keyboard and expression editor can be reached using the buttons shown above.

The various screens of PolyScope are described in the following sections.

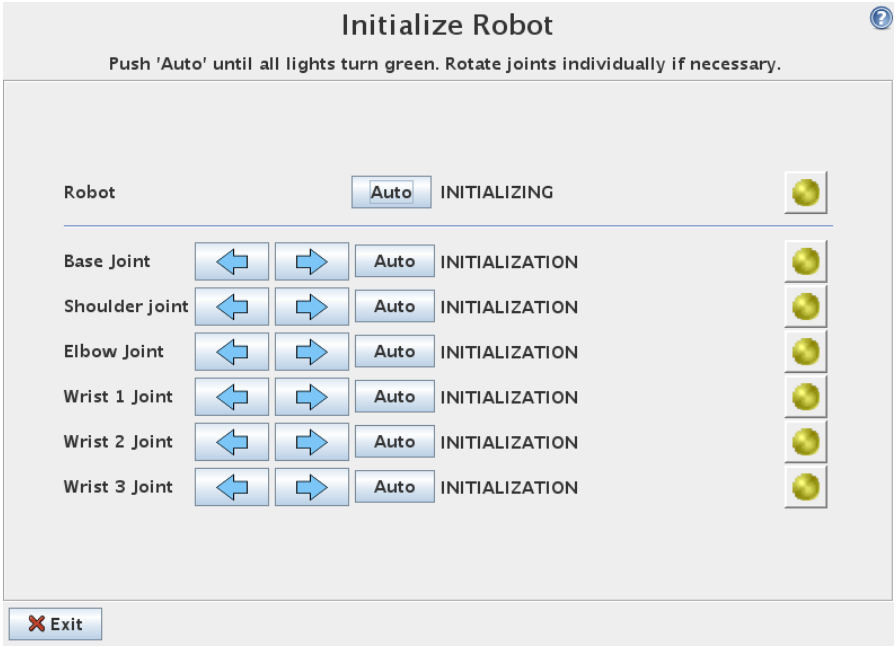
### 3.1.1 Welcome Screen




After booting up the controller PC, the welcome screen is shown. The screen offers the following options:

- **Run Program:** Choose a program to run. This is the simplest way to operate the robot, but requires a suitable program to have already been produced.
- **Program Robot:** Change a program, or create a new program.
- **Setup:** Set passwords, upgrade software via the Internet, request support, calibrate the touch screen, etc.
- **Shut Down Robot:** Shuts down the Controller PC and powers off the robot.

### 3.1.2 Initialization Screen



On this screen you control the initialization of the robot. When turned on, the robot needs to find the positions of each joint. To get the joint positions, the robot needs to move each joint.

#### Status LEDs

The status LEDs give an indication of the joints running state.

- A bright red LED tells that the robot is currently in a stopped state where the reasons can be several.
- A bright yellow LED indicates that the joint is running, but dosn't know its present position and needs homing.
- Finally a green LED indicates that the joint is running correctly and is ready to execute.

All the LEDs have to be green in order for the robot to operate normally.

#### Auto movement (Auto Buttons)

Normally it is always advisable to use the auto buttons to move the individual joints until they reach a known state. In order to operate the button, you have to press on the Auto button, and keep it pressed.

The auto buttons can be pressed individually for each joint, or for the whole robot. Great care should be taken if the robot is touching an obstacle or table, since driving the robot into the obstacle might damage a joint gearbox.

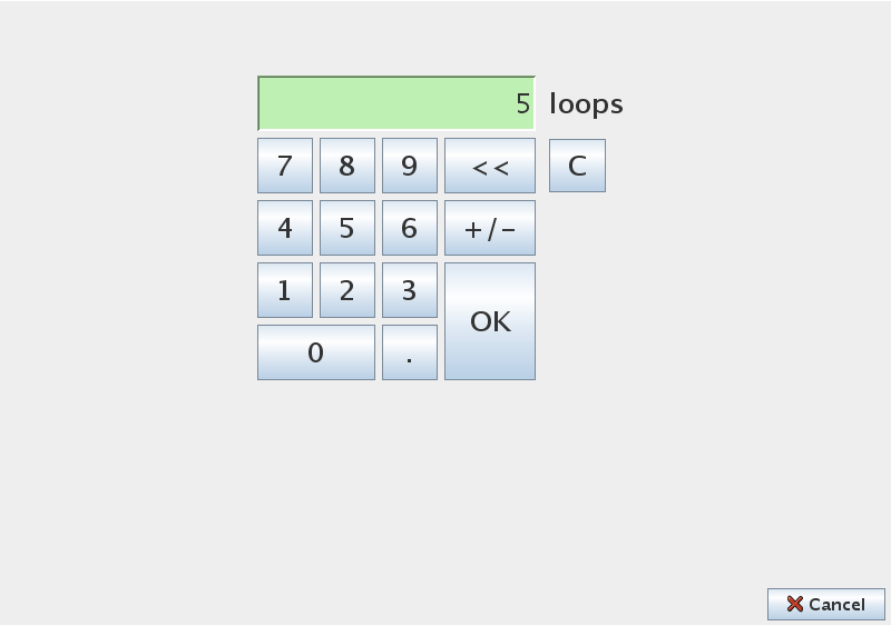
#### Moving directly (Move Buttons)

In the case where a joint is in a position where there is a major risk that uncontrolled motion would cause damage to the robot or its surroundings.

The operator can choose to home the robot manually for each joint. Each joint needs to move until the status LED changes to green (see section 3.1.2.

## 3.2 On-screen Editors

### 3.2.1 On-screen Keypad

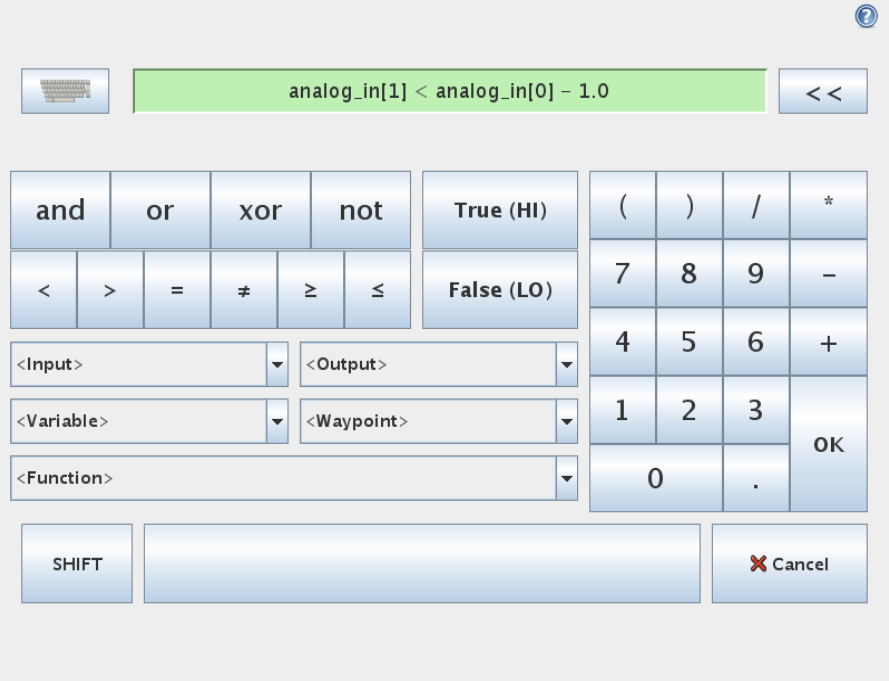



Simple number typing and editing facilities. In many cases, the unit of the typed value is displayed next to the number.

### 3.2.2 On-screen Keyboard

Simple text typing and editing facilities. The `Shift` key can be used to get some additional special characters.

### 3.2.3 On-screen Expression Editor




While the expression itself is edited as text, the expression editor has a number of buttons and functions for inserting the special expression symbols, such as $*$ for multiplication and $\leq$ for less than or equal to. The keyboard symbol button in the top right of the screen switches to text-editing of the expression. All defined variables can be found in the `Variable` selector, while the names of the input and output ports can be found in the `Input` and `Output` selectors. Some special functions are found in `Function`.

The expression is checked for grammatical errors when the `Ok` button is pressed. The `Cancel` button leaves the screen, discarding all changes.

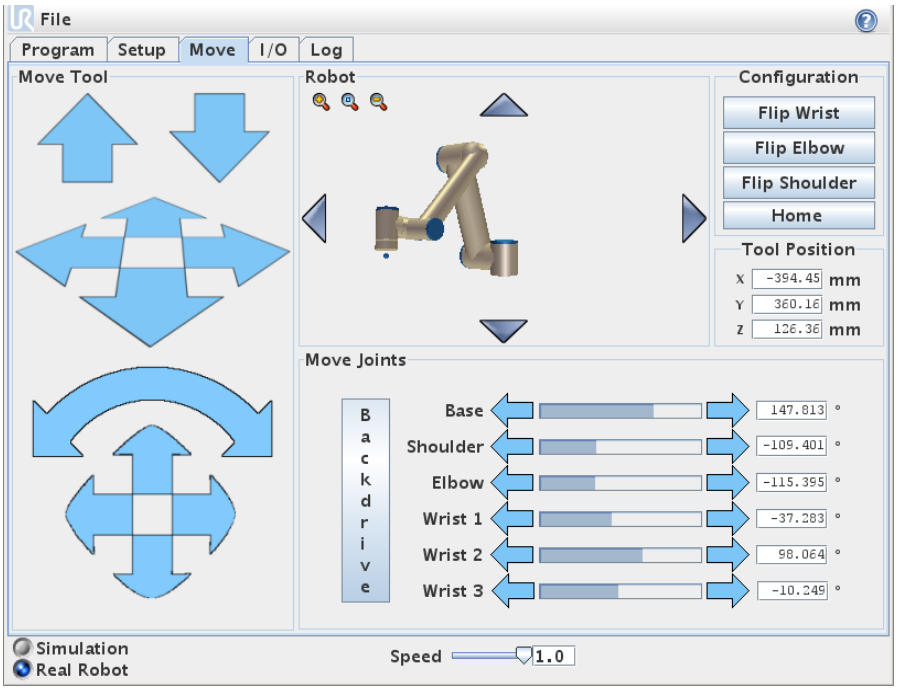
An expression can look like this:

```
digital_in[1]=True and analog_in[0]<0.5
```

## 3.3 Robot Control

### 3.3.1 Move Tab

On this screen you can always move (jog) the robot directly, either by translating/rotating the robot tool, or by moving robot joints individually.




### Robot

The current position of the robot is shown. Push the magnifying glass icons to zoom in/out and the arrow icons to translate or rotate the view. The viewing angle of the 3D drawing should match your view of the real robot.

### Move Tool

- Holding down a `translate arrow` (top) will move the tool-tip of the robot in the direction indicated.
- Holding down a `rotate arrow` (button) will change the orientation of the robot tool in the indicated direction. The point of rotation is the TCP, drawn as a small green ball.

Note: *Release the button to stop the motion at any time!*

### Move Joints

Allows the individual joints to be controlled directly. Each joint can move from $-360°$ to $+360°$, which are the *joint limits* illustrated by the horizontal bar for each joint. If a joint reaches its joint limit, it cannot be driven any further away from $0°$.

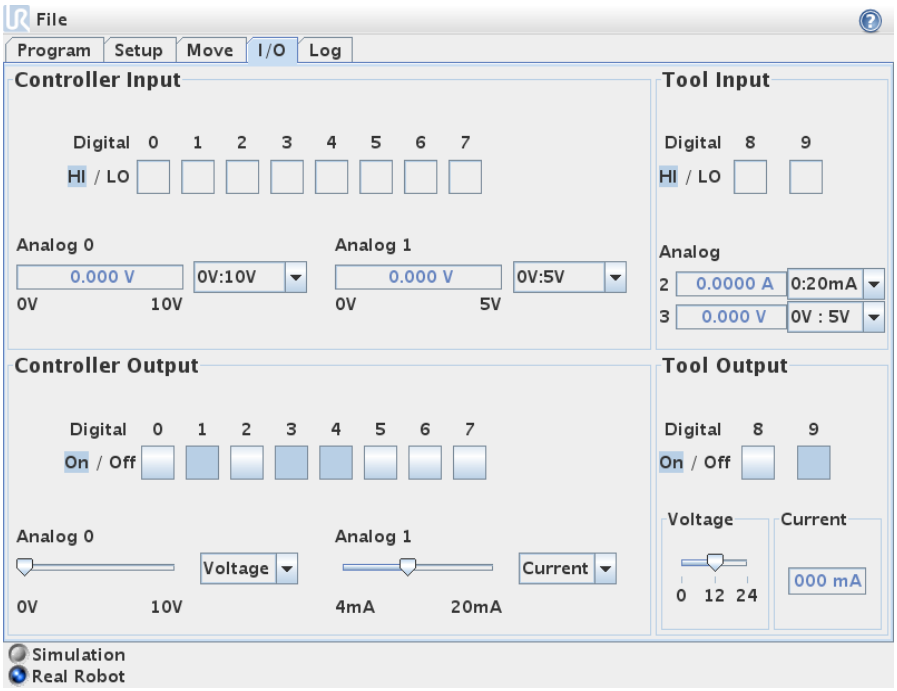
### Backdrive

While the 'Backdrive' button is held down, it is possible to physically grab the robot and pull it to where you want it to be. If the gravity setting (see 3.3.4) in the `Setup` tab is wrong, or the robot carries a heavy load, the robot might start moving (falling) when the 'Backdrive' button is pressed. In that case, just release the 'Backdrive' button again.

### Configuration

With these buttons you can change the joint position in such a way that the tool of the robot does not change position, but the robot arm changes side. Beware of collisions when using this feature.

## 3.3.2 I/O Tab



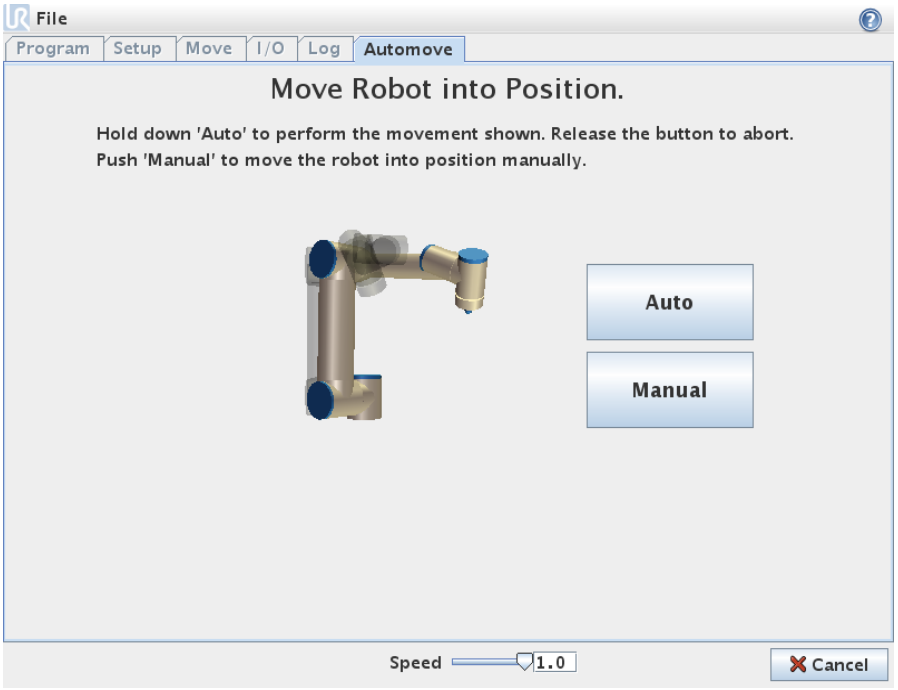

On this screen you can always monitor and set the I/O signals from/to the robot. The screen displays the current state of the I/O, inluding during program execution. If anything is changed during program execution, the program will stop. At program stop, all output signals will retain their states. The screen is updated at only 10Hz, so a very fast signal might not display properly.

The electrical details of the signals are described in section 2.1.

**Analog Range Settings** The analog output can be set to either current (4-20mA) or voltage (0-10V) output. The analog input ranges adjusted to be from (-10-10V) to (0-5V). The settings will be remembered for eventual later restarts of the robot controller when a program is saved.

### 3.3.3 AutoMove Tab

The AutoMove tab is used when the robot has to move to a specific position in its workspace. Examples are when the robot has to move to the start position of a program before running it, or when moving to a waypoint while modifying a program.




#### Animation

The animation shows the movement the robot is about to perform. Compare the animation with the position of the real robot and make sure that robot can safely perform the movement without hitting any obstacles.

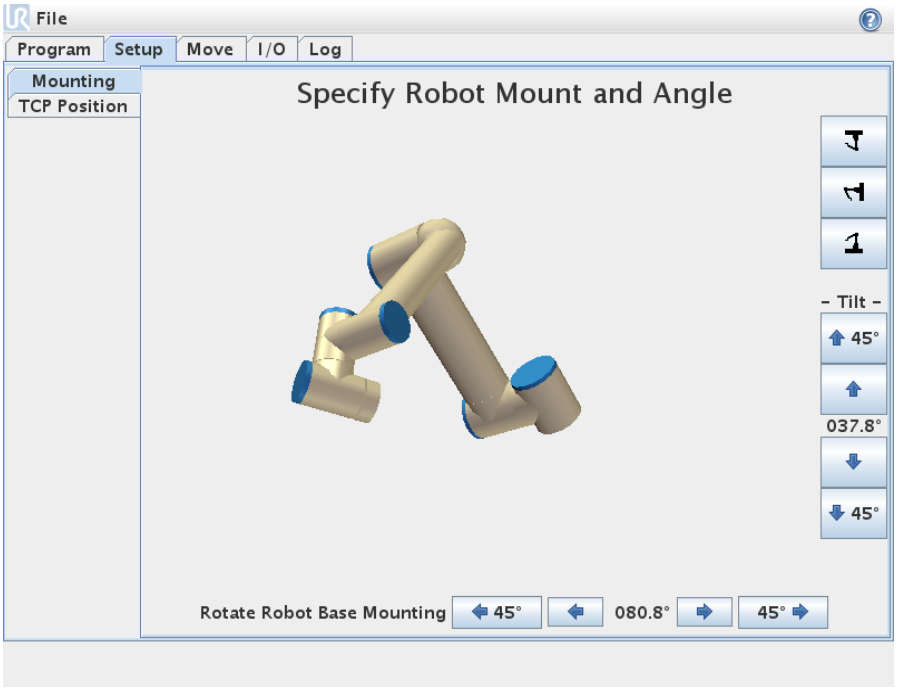
#### Auto

Hold down the `Auto` button to move the robot as shown in the animation. Note: *Release the button to stop the motion at any time!*

#### Manual

Pushing the `Manual` button will take you to the `MoveTab` where the robot can be moved manually. This is only needed if the movement in the animation is not preferable.

### 3.3.4 Setup → Mounting




Here the mounting of the robot can be specified. This serves two purposes:

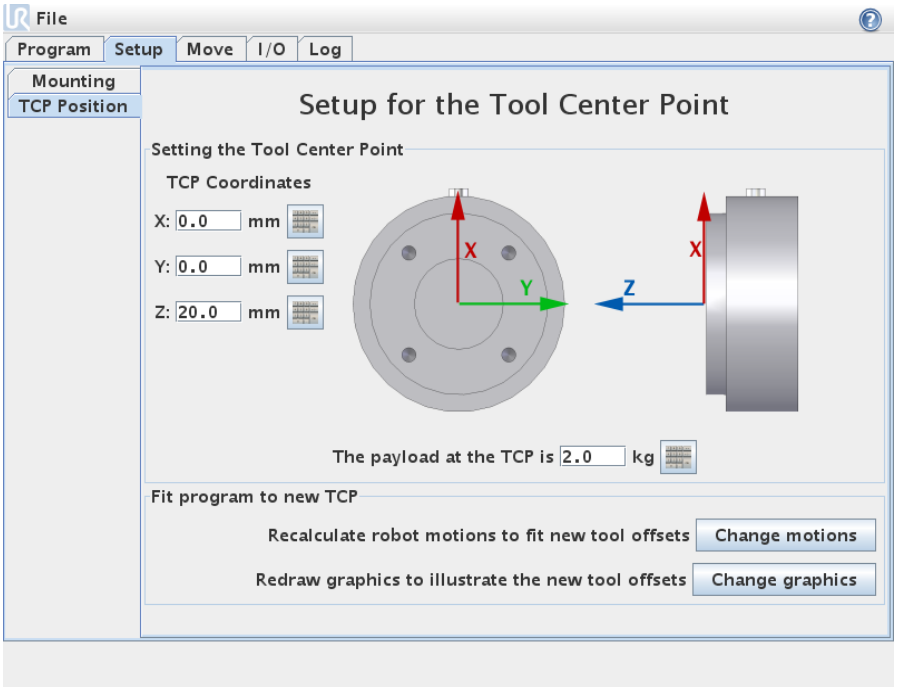
1. Making the robot look right on the screen.

2. Telling the controller about the direction of gravity.

The controller uses an advanced dynamics model to give the robot smooth and precise motions, and to make the robot hold itself when backdriven. For this reason, it is important that the mounting of the robot is set correctly.

The default is that the robot is mounted on a flat table or floor, in which case no change is needed on this screen. However, if the robot is *ceiling mounted*, *wall mounted* or mounted at an angle this can be adjusted using the push-buttons. The buttons on the right side of the screen are for setting the angle of the robot's mounting. The three top right side buttons set the angle to *ceiling* ($180°$), *wall* ($90°$), *floor* ($0°$). The `Tilt` buttons can be used to set an arbitrary angle. The buttons on the lower part of the screen are used to rotate the mounting of the robot to match the actual mounting.

### 3.3.5 Setup → TCP Position



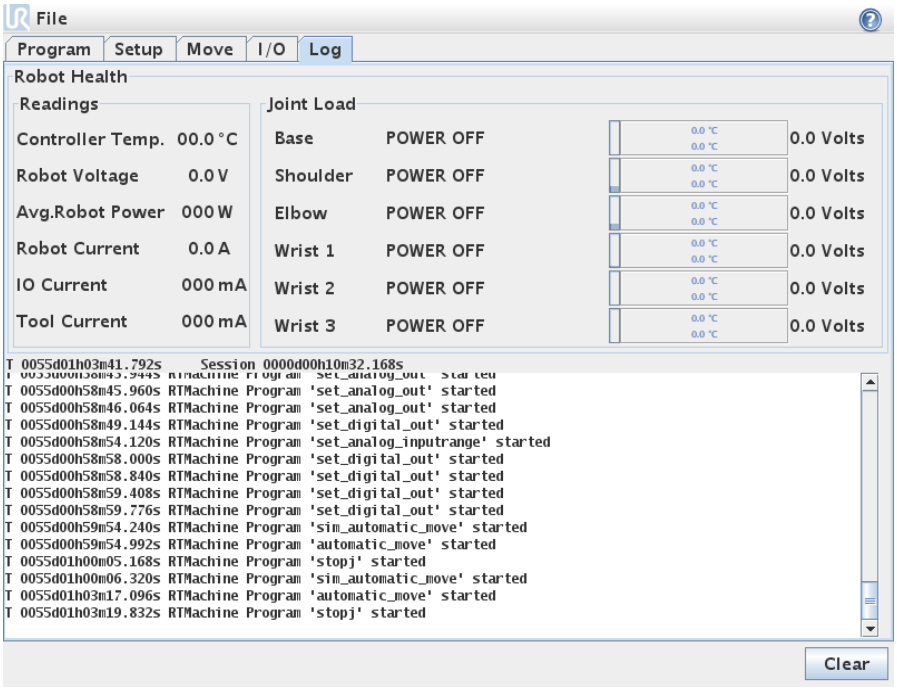

The *Tool Center Point* (TCP) is the characteristic point on the robot's tool. When the robot moves linearly, it is this point that moves in a straight line. It is also the motion of the TCP that is visualized on the graphics tab. The TCP is given relative to the center of the tool output flange, as indicated on the on-screen graphics.

The two buttons on the bottom of the screen are relevant when the TCP is changed.

- **Change Motions** recalculates all positions in the robot program to fit the new TCP. This is relevant when the shape or size of the tools has been changed.

- **Change Graphics** redraws the graphics of the program to fit the new TCP. This is relevant when the TCP has been changed without any physical changes to the tool.

### 3.3.6 Log Tab


File
Program
Setup
Move
I/O
Log
Robot Health
Readings
Controller Temp. 00.0 °C
Robot Voltage 0.0 V
Avg.Robot Power 000 W
Robot Current 0.0 A
IO Current 000 mA
Tool Current 000 mA
Joint Load
Base POWER OFF 0.0 Volts
Shoulder POWER OFF 0.0 Volts
Elbow POWER OFF 0.0 Volts
Wrist 1 POWER OFF 0.0 Volts
Wrist 2 POWER OFF 0.0 Volts
Wrist 3 POWER OFF 0.0 Volts
T 0055d01h03m41.792s Session 0000d00h10m32.168s
T 0055d00h58m45.960s RTMachine Program 'set_analog_out' started
T 0055d00h58m46.064s RTMachine Program 'set_analog_out' started
T 0055d00h58m49.144s RTMachine Program 'set_digital_out' started
T 0055d00h58m54.120s RTMachine Program 'set_analog_inputrange' started
T 0055d00h58m58.000s RTMachine Program 'set_digital_out' started
T 0055d00h58m58.840s RTMachine Program 'set_digital_out' started
T 0055d00h58m59.408s RTMachine Program 'set_digital_out' started
T 0055d00h58m59.776s RTMachine Program 'set_digital_out' started
T 0055d00h59m54.240s RTMachine Program 'sim_automatic_move' started
T 0055d00h59m54.992s RTMachine Program 'automatic_move' started
T 0055d01h00m05.168s RTMachine Program 'stopj' started
T 0055d01h00m06.320s RTMachine Program 'sim_automatic_move' started
T 0055d01h03m17.096s RTMachine Program 'automatic_move' started
T 0055d01h03m19.832s RTMachine Program 'stopj' started
Clear

**Robot Health** The top half of the screen displays the health of the robot. The left part shows information related to the control box of the robot, while the right part shows information about each robot joint. Each robot joint shows information for temperaure of the motor and electronics, the load of the joint and the voltage at the joint.

**Robot Log** On the bottom half of the screen log messages are shown. The first column shows the time of arrival of the message. The next column shows the sender of the message. The last column shows the message itself.

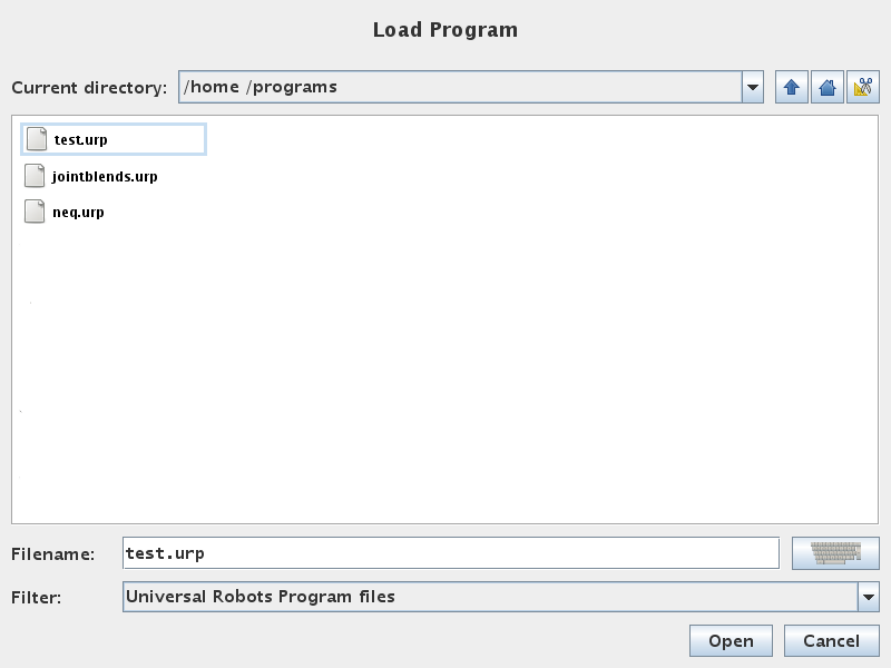
### 3.3.7 Load Screen

On this screen you choose which program to load. There are two versions of this screen: one that is to be used when you just want to load a program and execute it, and one that is used when you want to actually select and edit a files program.

The main difference lies in which actions are available to the user. In the basic load screen, the user will only be able to access files - not modify or delete them. Furthermore, the user is not allowed to leave the directory structure that descends from the `programs` folder. The user can descend to a sub-directory, but he cannot get any higher than the `programs` folder.

Therefore, all programs should be placed in the programs folder and/or sub folders under the programs folder.

### Screen layout




This image shows the actual load screen. It consists of the following important areas and buttons.

**Path history** The path history shows a list of the paths leading up to the present location. This means that all parent directories up to the root of the computer are shown. Here you must notice that you may not be able to access all the directories above the programs folder.

By selecting a folder name in the list, the load dialog changes to that directory and displays it in the file selection area 3.3.7.

**File selection area** In this area of the dialog the contents of the actual area is present. It gives the user the option to select a file by single clicking on its name or to open the file by double clicking on its name.

In the case that the user double-clicks on a directory, the dialog descends into this folder and presents its contents.

**File filter** By using the file filter, one can limit the files shown to include the type of files that one wishes. By selecting "Backup Files" the file selection area will display the latest 10 saved versions of each program, where `.old0` is the newest and `.old9` is the oldest.

**File field** Here the currently selected file is shown. The user has the option to manually enter the file name of a file by clicking on the keyboard icon to the right of the field. This will cause an on-screen keyboard to pop up where the user can enter the file name directly on the screen.

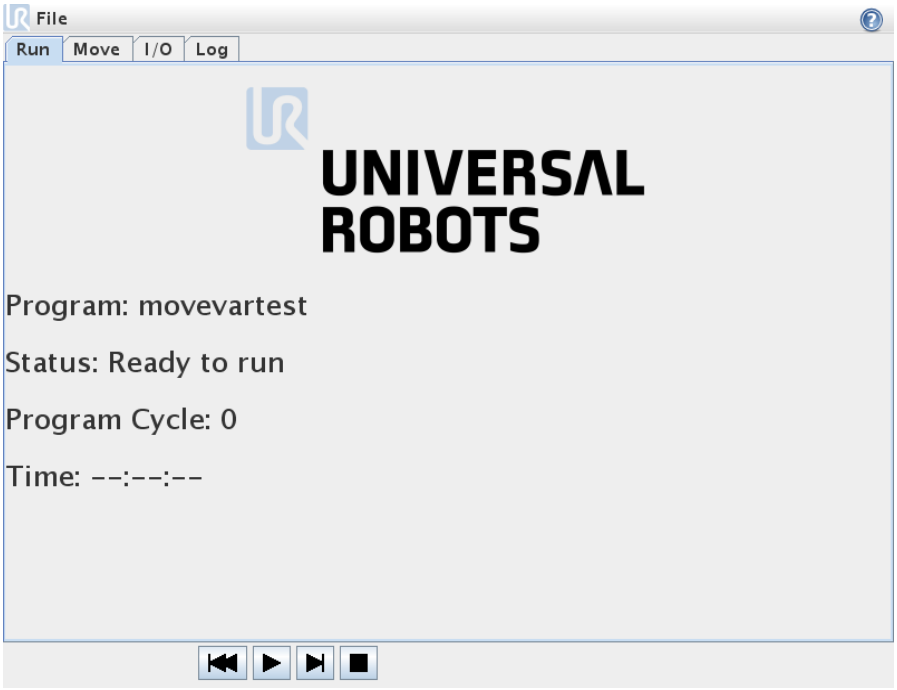
**Open button** Clicking on the Open button, will open the currently selected file and return to the previous screen.

**Cancel button** Clicking on the Cancel button will abort the current loading process and cause the screen to switch to the previous image.

**Action buttons** A series of buttons gives the user the ability to perform some of the actions that normally would be accessible by right-clicking on a file name in a conventional file dialog. Added to this is the ability to move up in the directory structure and directly to the program folder.

- Parent: Move up in the directory structure. The button will not be enabled in two cases: when the current directory is the top directory or if the screen is in the limited mode and the current directory is the program folder.
- Go to program folder: Go home
- Actions: Actions such as create directory, delete file etc.

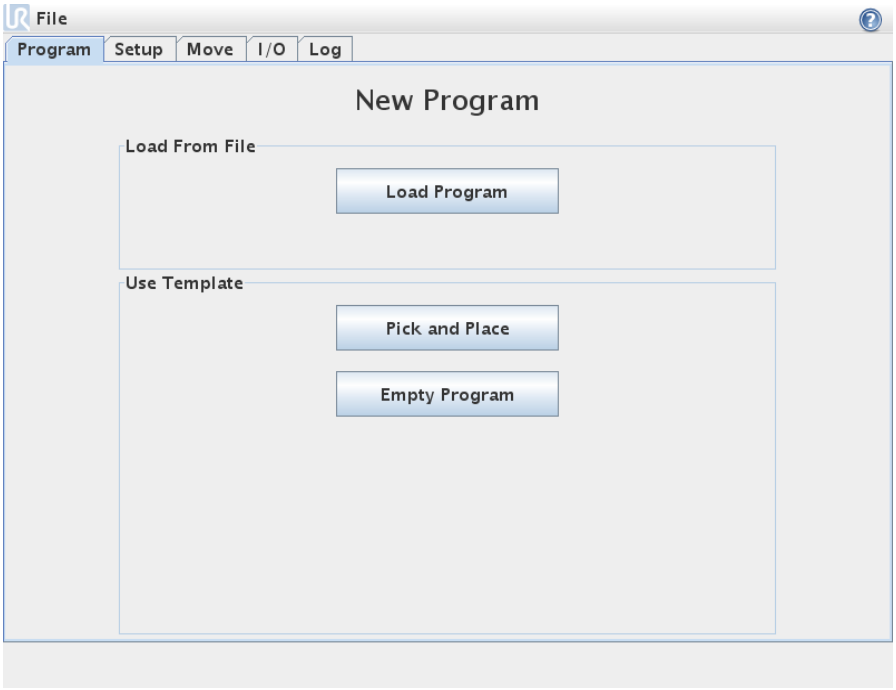
### 3.3.8 Run Tab




This tab provides a very simple way of operating the robot, with as few buttons and options as possible. This can be useful combined with password protecting the programming part of PolyScope (see section 3.5.4), to make the robot into a tool that can run exclusively pre-written programs.

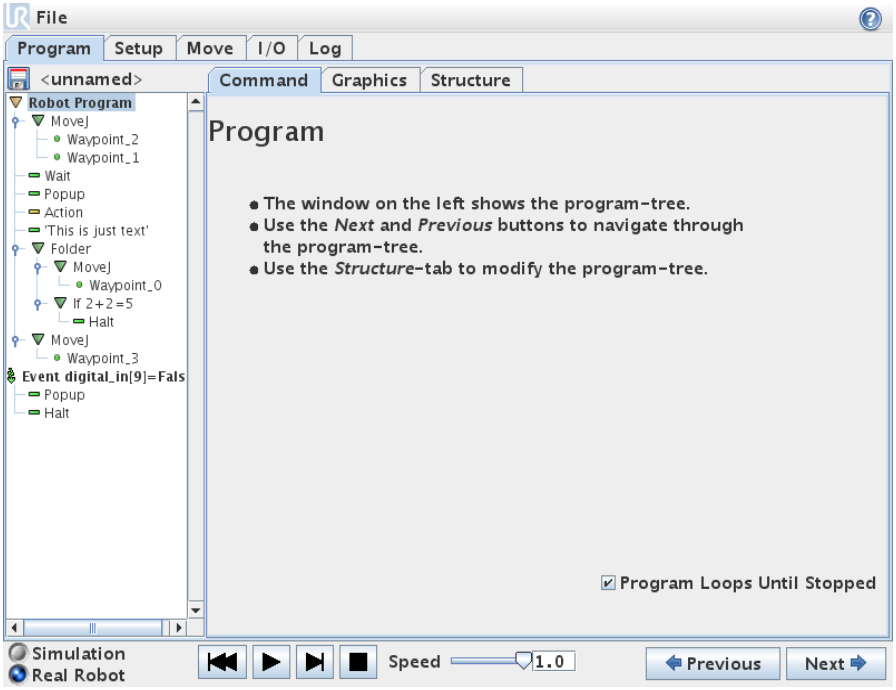
## 3.4 Programming

### 3.4.1 Program → New Program




A new robot program can start from either a *template* or from an existing (saved) robot program. A *template* can provide the overall program structure, so only the details of the program need to be filled in.

### 3.4.2 Program Tab




The program tab shows the current program being edited.

The *program tree* on the left side of the screen displays the program as a list of commands, while the area on the right side of the screen displays information relating to the current command. The current command is selected by clicking the command list, or by using the `Previous` and `Next` buttons on the

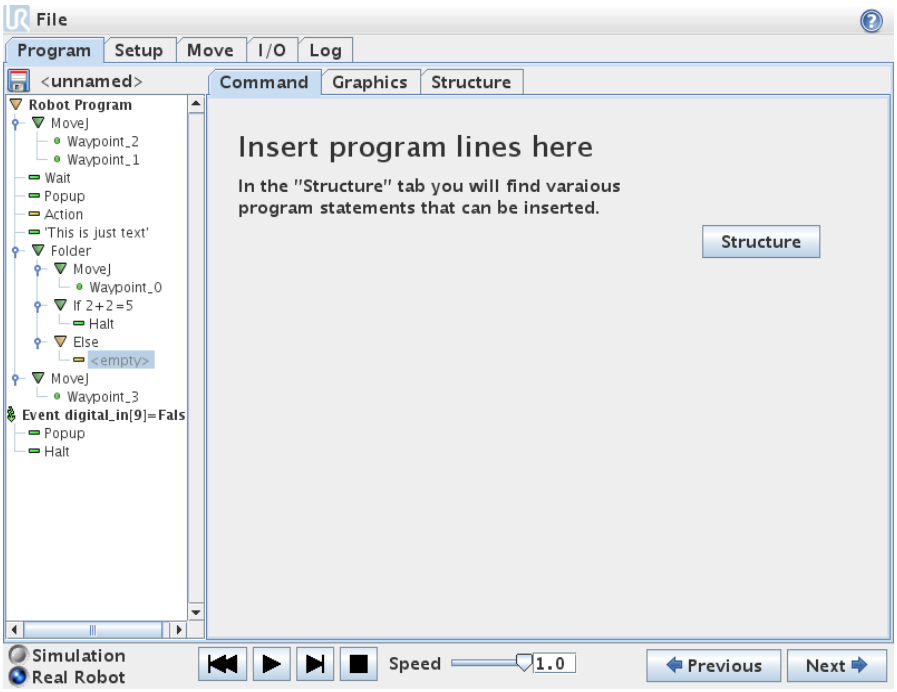
bottom right of the screen. Commands can be inserted or removed using the `Structure` tab, described in section 3.4.25. The program name is shown directly above the command list, with a small disk icon that can be clicked to quickly save the program.

The lowest part of the screen is the *Dashboard*. The *Dashboard* features a set of buttons similar to an old-fashioned tape recorder, from which programs can be started and stopped, single-stepped and restarted. The *speed slider* allow you to adjust the program speed at any time, which directly affects the speed at which the robot moves. To the left of the *Dashboard* the `Simulation` and `Real Robot` buttons toggle between running the program in a simulation, or running it on the real robot. When running in simulation, the robot does not move and thus cannot damage itself or any nearby equipment in collisions. Use simulation to test programs if unsure about what the robot will do.

While the program is being written, the resulting motion of the robot is illustrated using a 3D drawing on the `Graphics` tab, described in section 3.4.24.

Next to each program command is a small icon, which is either red, yellow or green. A red icon means that there is an error in that command, yellow means that the command is not finished, and green means that all is OK. A program can only be run when all commands are green.

### 3.4.3 Program → Command Tab, <Empty>




Program commands need to be inserted here. Press the 'Structure' button to go to the structure tab, where the various selectable program lines can be found. A program cannot run before all lines are specified and defined.

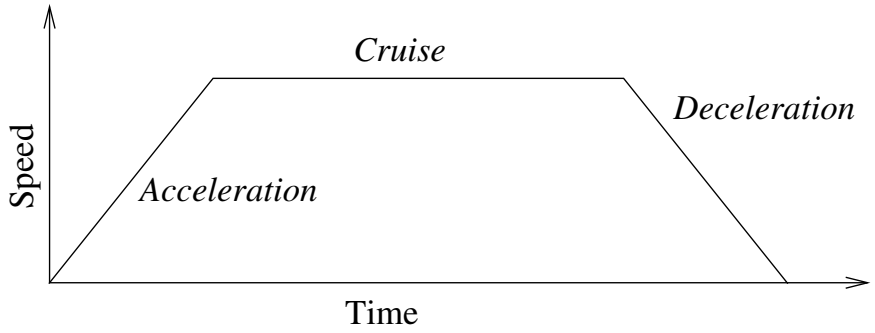
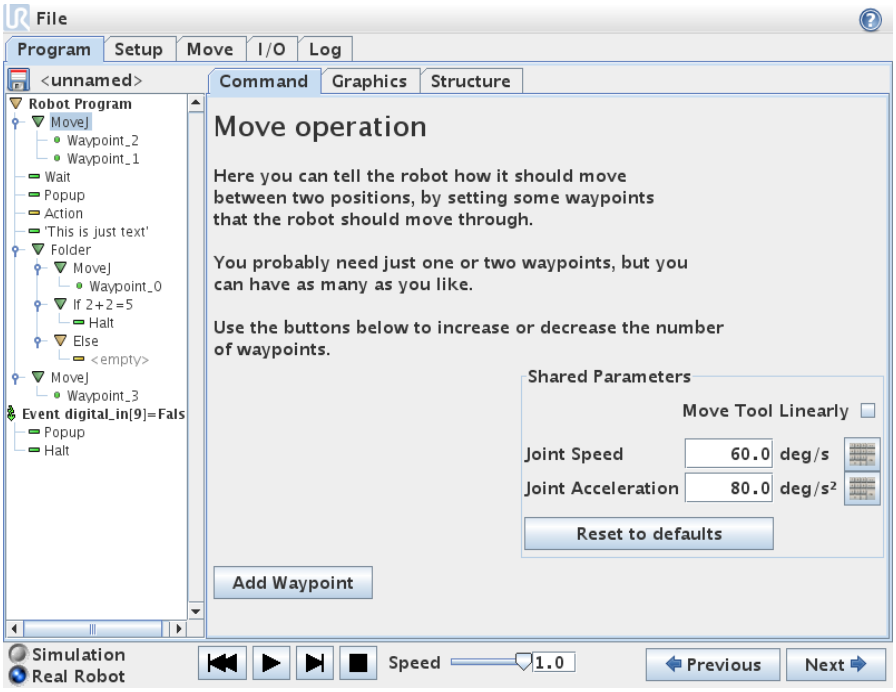



Figure 3.1: Speed profile for a motion. The curve is divided into three segments: *acceleration*, *cruise* and *deceleration*. The level of the *cruise* phase is given by the speed setting of the motion, while the steepness of the *acceleration* and *deceleration* phases is given by the acceleration parameter.

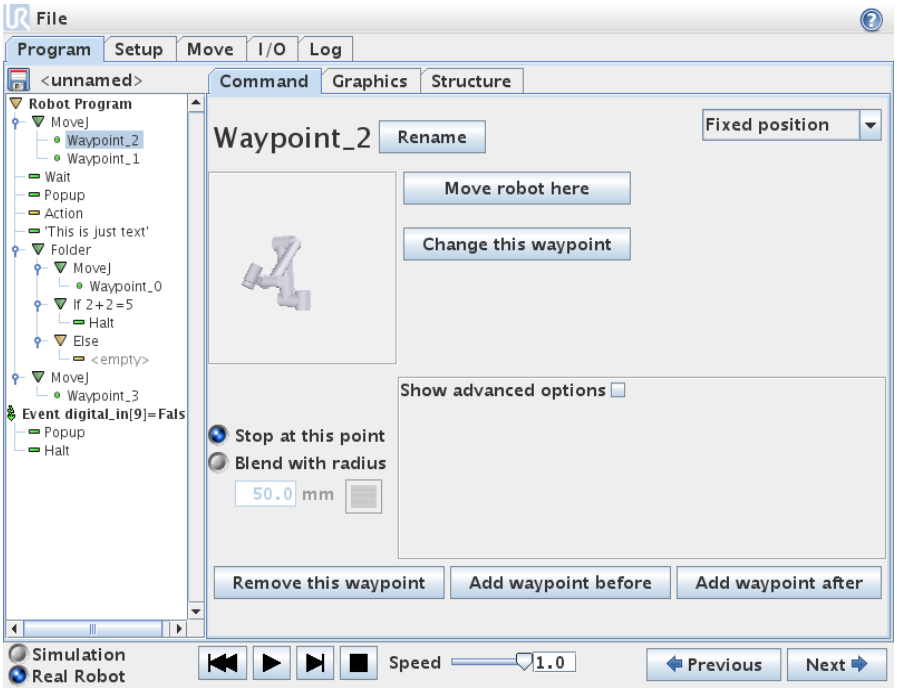
### 3.4.4 Program → Command Tab, Move




The Move command controls the robot motion through the underlying waypoints. Waypoints have to be under a Move command. The Move command defines the acceleration and the speed at which the robot is moving, and also whether the motion is in *joint space* or *linear space*. In *joint space* each joint is controlled to reach the desired end location at the same time, which results in a curved path for the tool, whereas in *linear space* the joints perform a more complicated motion to keep the tool on a straight line path. Generally, the robot can move faster in *joint space*. In the program tree view, the command will switch between `movej` and `movel` to display what type of motion is selected.

The settings of a Move command apply to the path from the robot's current position to the first waypoint under the command, and from there to each of the following waypoints. The Move command settings do not apply to the path going *from* the last waypoint under that Move command.

### 3.4.5 Program → Command Tab, Fixed Waypoint




A point on the robot path. Waypoints are the most central part of a robot program, telling the robot where to be. A fixed position waypoint is given by physically moving the robot to the position.

#### Waypoint names

Waypoint names can be changed. Two waypoints with the same name is always the same waypoint. Waypoints are numbered as they are specified.

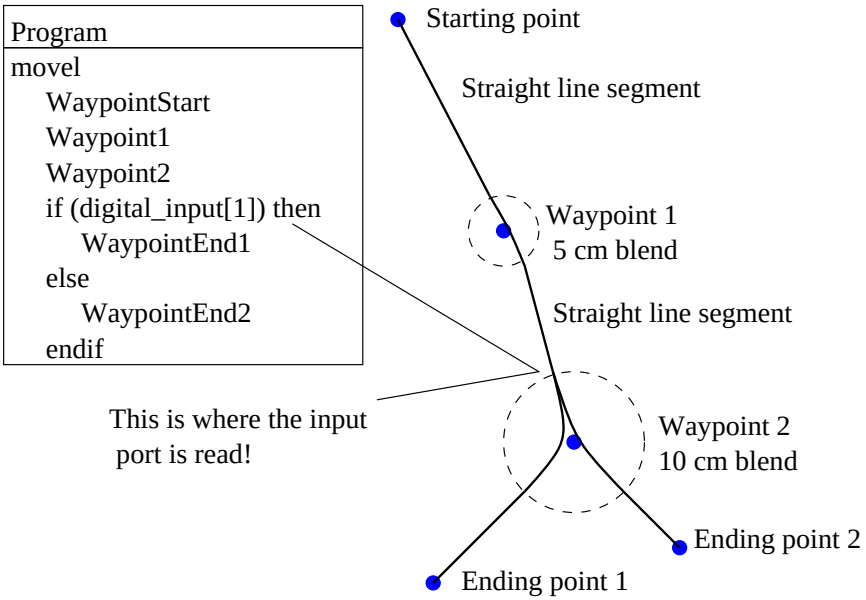
#### Blend radius

If a blend radius is set, the robot trajectory blends around the waypoint, allowing the robot not to stop at the point. Blends cannot overlap, so it is not possible to set a blend radius that overlaps a blend radius for a previous or following waypont. A stop point is a waypoint with a blend radius of $0.0mm$.

#### Note on I/O Timing

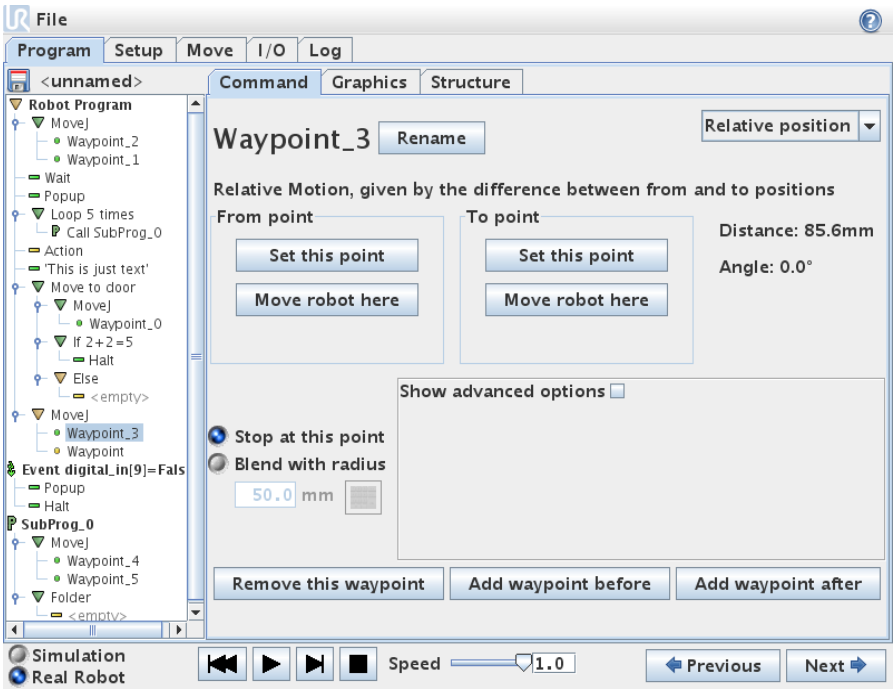
If a waypoint is a stop point with an I/O command as the next command, the I/O command is executed when the robot stops at the waypoint. However, if the waypoint has a blend radius, the following I/O command is executed when the robot enters the blend.

**Example**




A small example in which a robot program moves the tool from a starting position to one of two ending positions, depending on the state of `digital_input[1]`. Notice that the tool trajectory (thick black line) moves in straight lines outside the blend areas (dashed circles), while the tool trajectory deviates from the straight line path inside the blend areas. Also notice that the state of the `digital_input[1]` sensor is read just as the robot is about to enter the blend area around `Waypoint 2`, even though the `if...then` command is after `Waypoint 2` in the program sequence. This is somewhat counter-intuitive, but is necessary to allow the robot to select the right blend path.

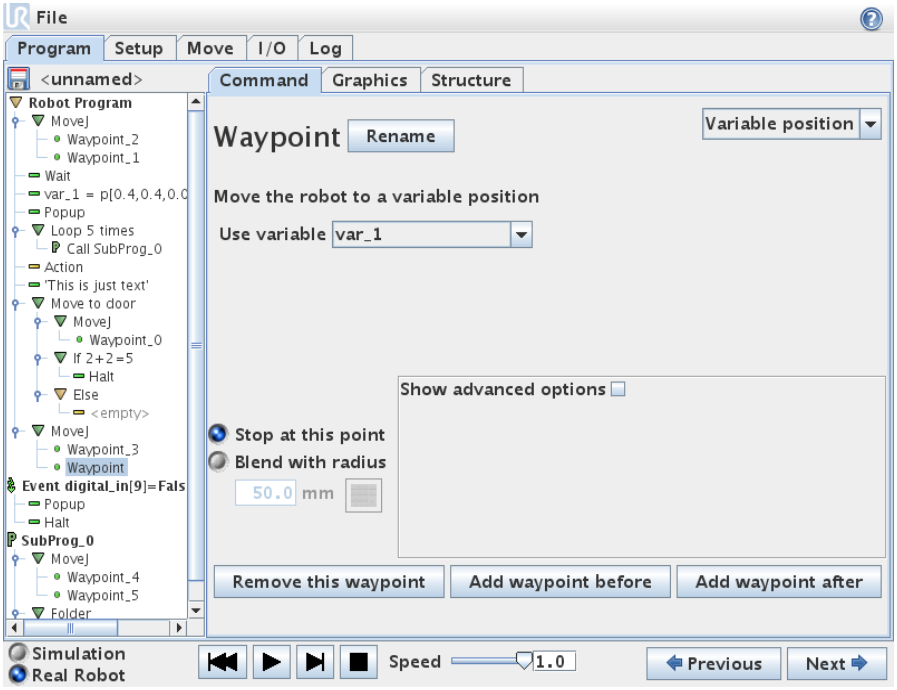
### 3.4.6 Program → Command Tab, Relative Waypoint




A waypoint with the position given relative to the robot's previous position, such as "two centimeters to the left". The relative position is defined as the

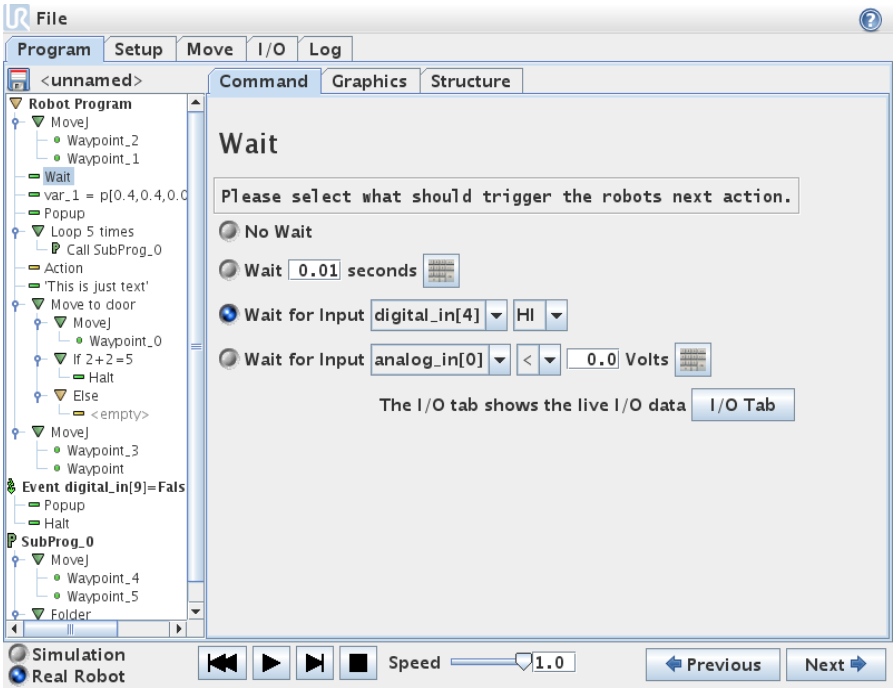
difference between the two given positions (left to right). Note that repeated relative positions can move the robot out of its workspace.

### 3.4.7 Program → Command Tab, Variable Waypoint



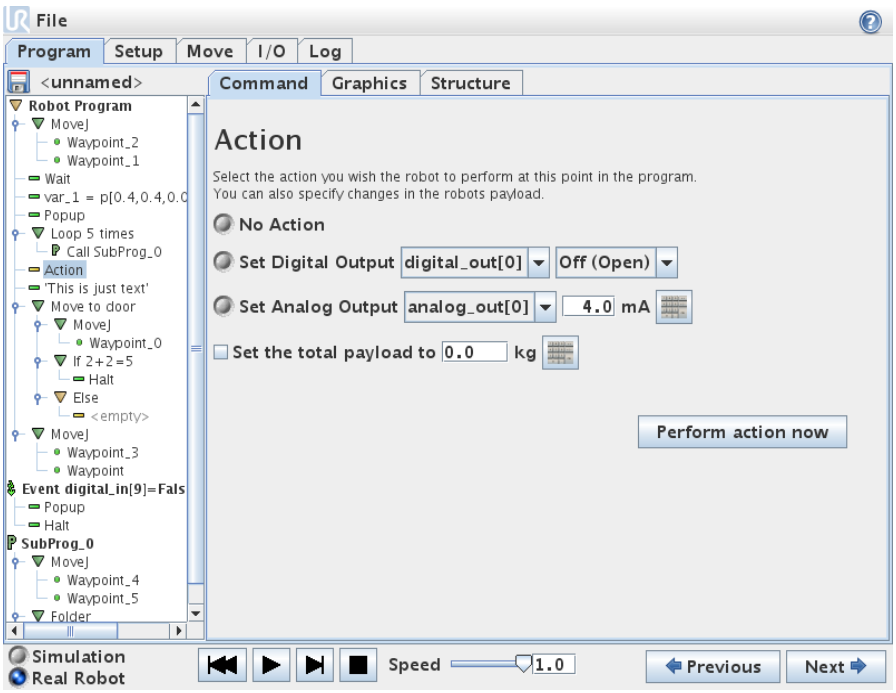

A waypoint with the position given by a variable, in this case `calculated_pos`. The variable can be a list of joint angles in radians, such as given by the assignment `var=[0.1,0.4,0.2,2.0,2.1,-3.14]`, or a *pose* such as `var=p[0.5,0.0,0.0,3.14,0.0,0.0]`. The first three are *x,y,z* and the last three are the orientation given as an *axis-angle* given by the vector *rx,ry,rz*. The length of the axis is the angle to be rotated in radians, and the vector itself gives the axis about which to rotate.

### 3.4.8 Program → Command Tab, Wait



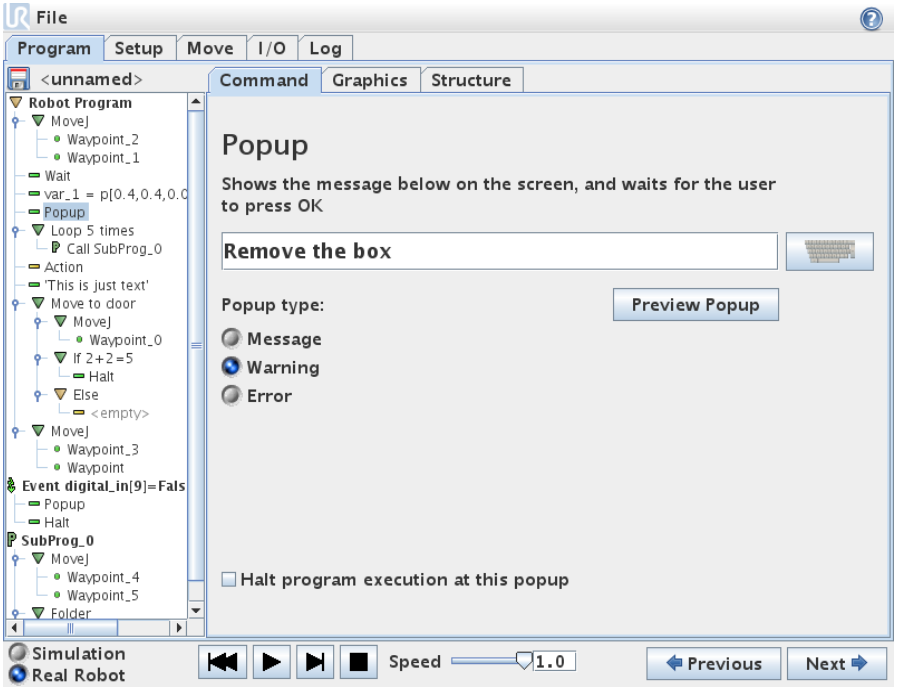

Waits for a given amount of time or for an I/O signal.

### 3.4.9 Program → Command Tab, Action



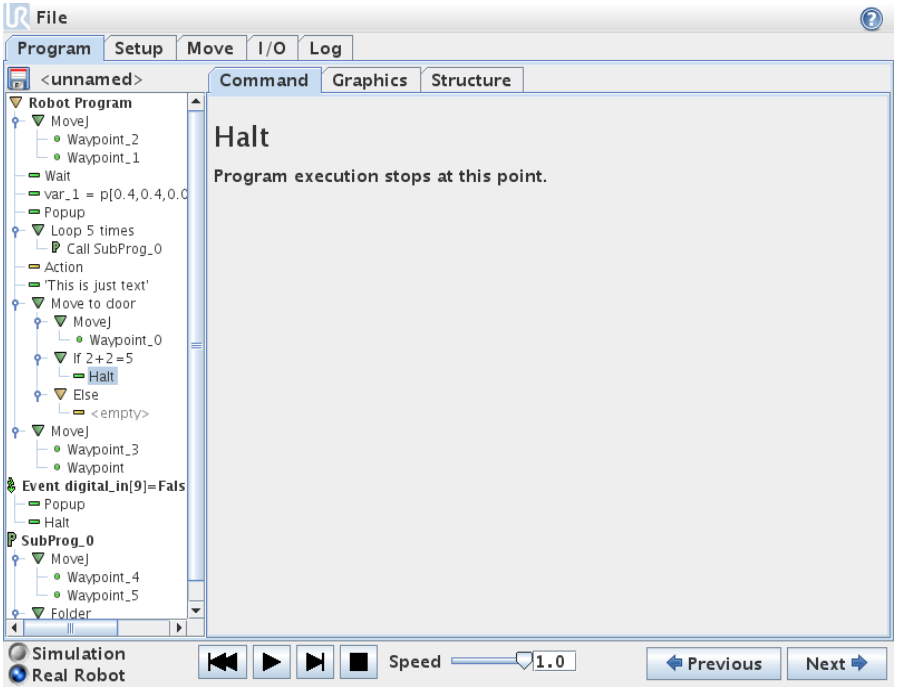

Sets either digital or analog outputs to a given value. Can also be used to set the payload of the robot, for example the weight that is picked up as a consequence of this action. Adjusting the weight can be neccesary to prevent the robot from security stopping unexpectedly, when the weight at the tool is different to that which is excpected.

### 3.4.10 Program → Command Tab, Popup



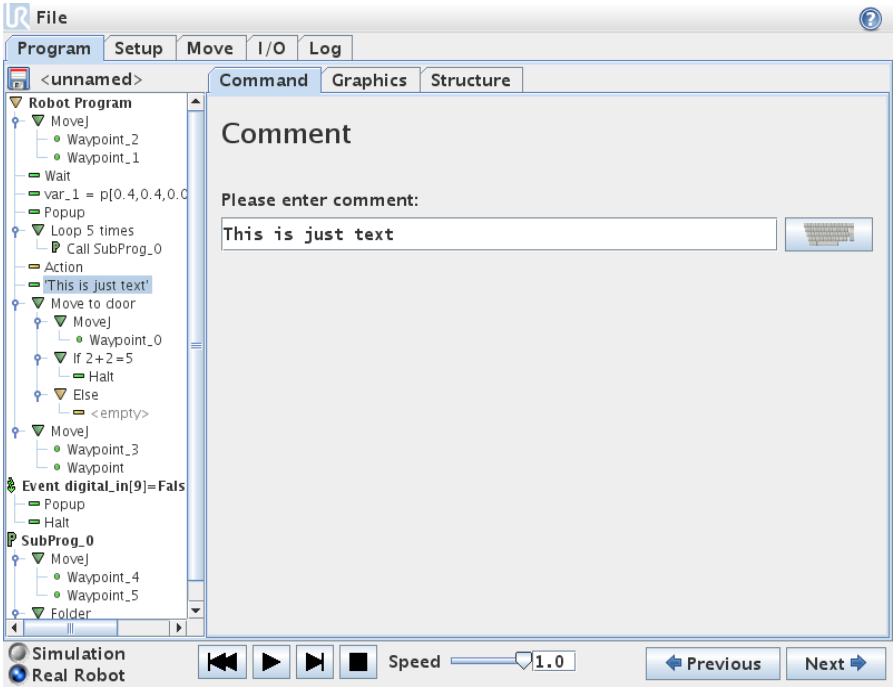

The popup is a message that appears on the screen when the program reaches this command. The style of the message can be selected, and the text itself can be given using the on-screen keyboard. The robot waits for the user/operator to press the "OK" button under the popup before continuing the program. If the "Halt program execution" item is selected, the robot program halts at this popup.

### 3.4.11 Program → Command Tab, Halt



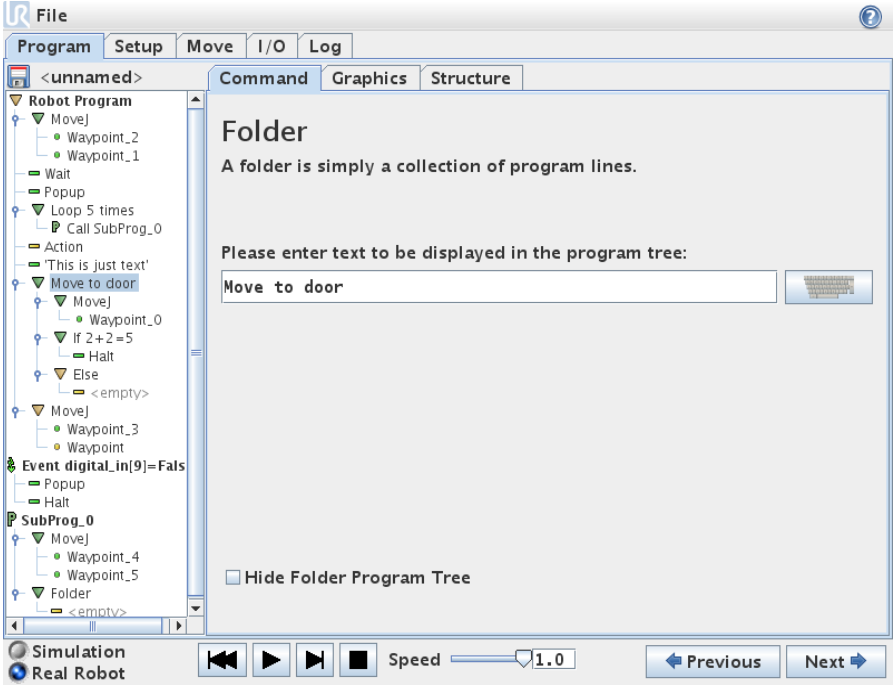

The program execution stops at this point.

### 3.4.12 Program → Command Tab, Comment




Gives the programmer an option to add a line of text to the program. This line of text does not do anything during program execution.

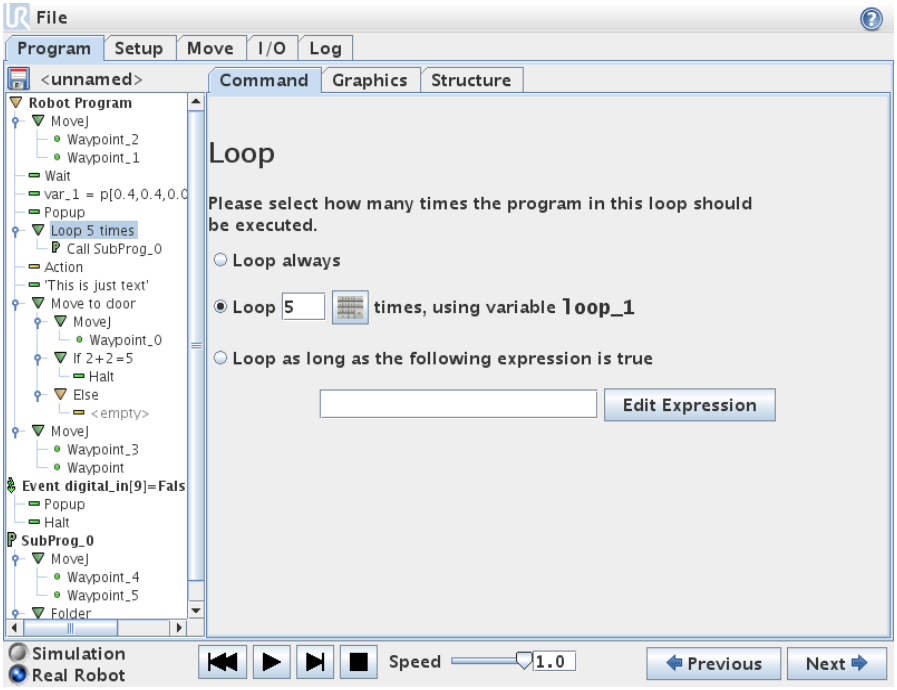
### 3.4.13 Program → Command Tab, Folder




A folder is used to organize and label specific parts of a program, to clean up the program tree, and to make the program easier to read and navigate.

A folder does not in itself do anything.

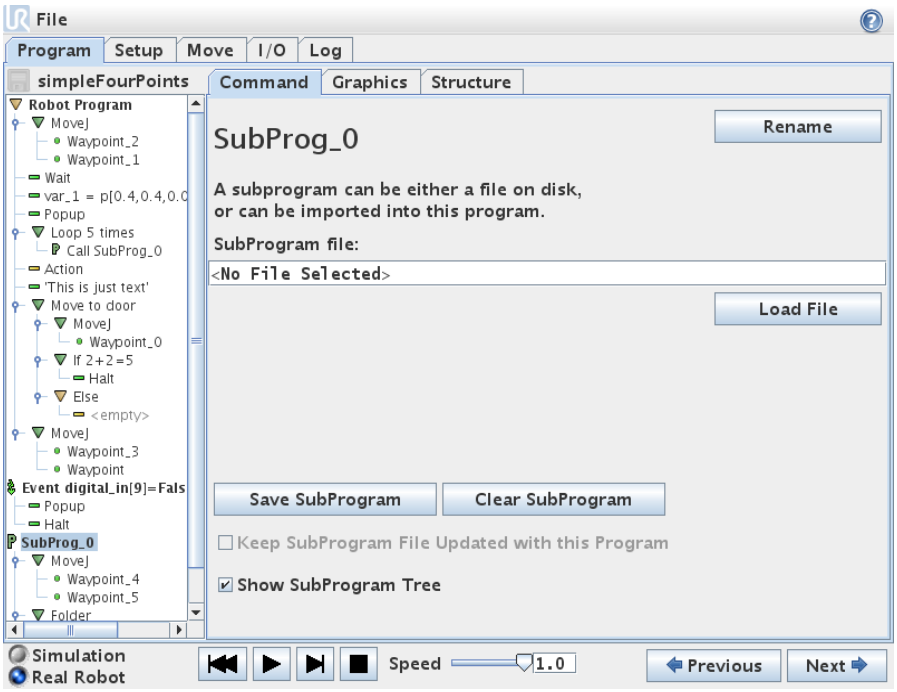
### 3.4.14 Program → Command Tab, Loop




Loops the underlying program commands. Depending on the selection, the underlying program commands are either looped infinitely, a certain number of times or as long as the given condition is true. When looping a certain number of times, a dedicated loop variable (called `loop_1` in the screen shot above) is created, which can be used in expressions within the loop. The loop variable counts from $0$ to $N - 1$.

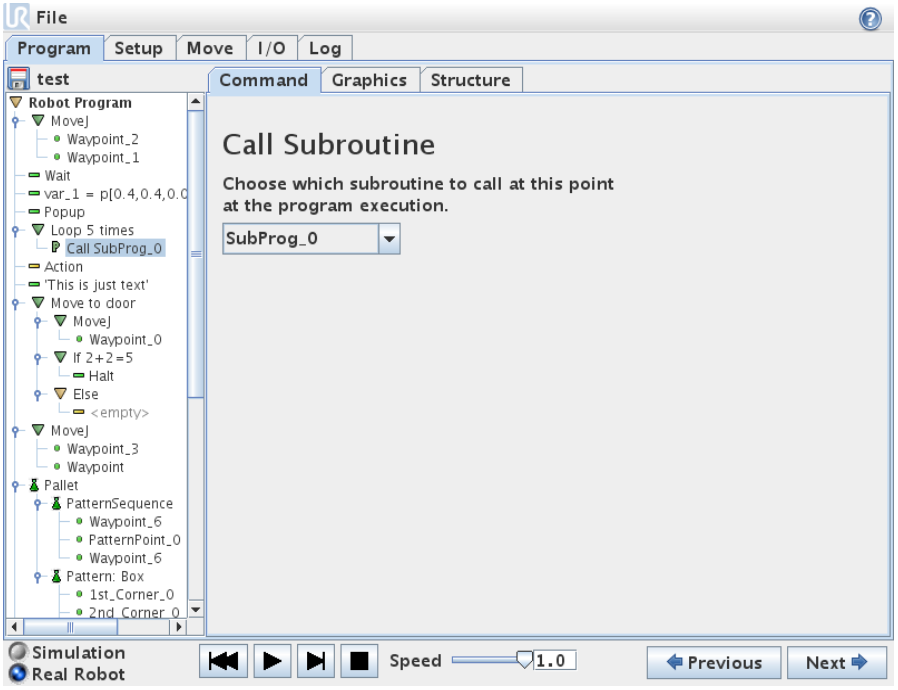
When looping using an expression as end condition, PolyScope provides an option for continuously evaluating that expression, so that the "loop" can be interrupted anytime during its execution, rather that just after each iteration.

### 3.4.15 Program → Command Tab, SubProgram



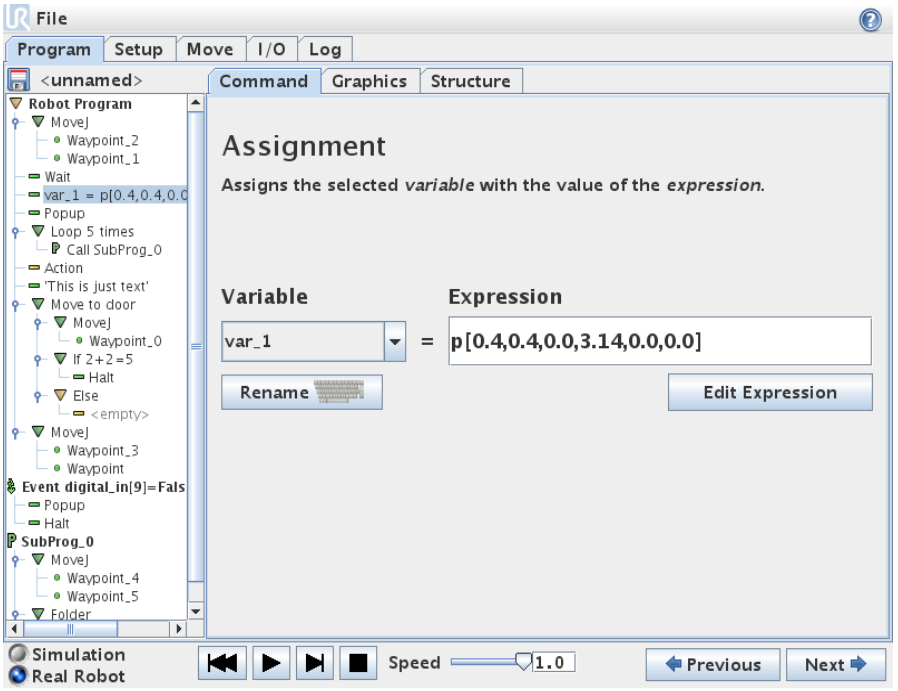
A Sub Program can hold program parts that are needed several places. A Sub Program can be a seperate file on the disk, and can also be hidden to protect against accidental changes to the SubProgram.

**Program → Command Tab, Call SubProgram**




A call to a sub program will run the program lines in the sub program, and then return to the following line.

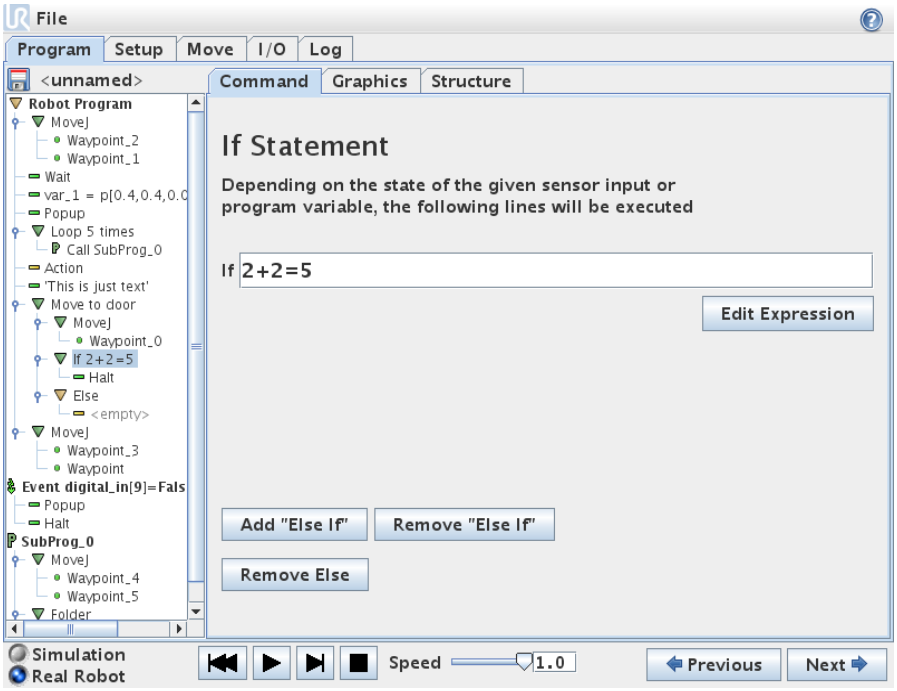
### 3.4.16 Program → Command Tab, Assignment




Assigns values to variables. An assignment puts the computed value of the right hand side into the variable on the left hand side. This can be useful in

complex programs.

### 3.4.17 Program → Command Tab, If

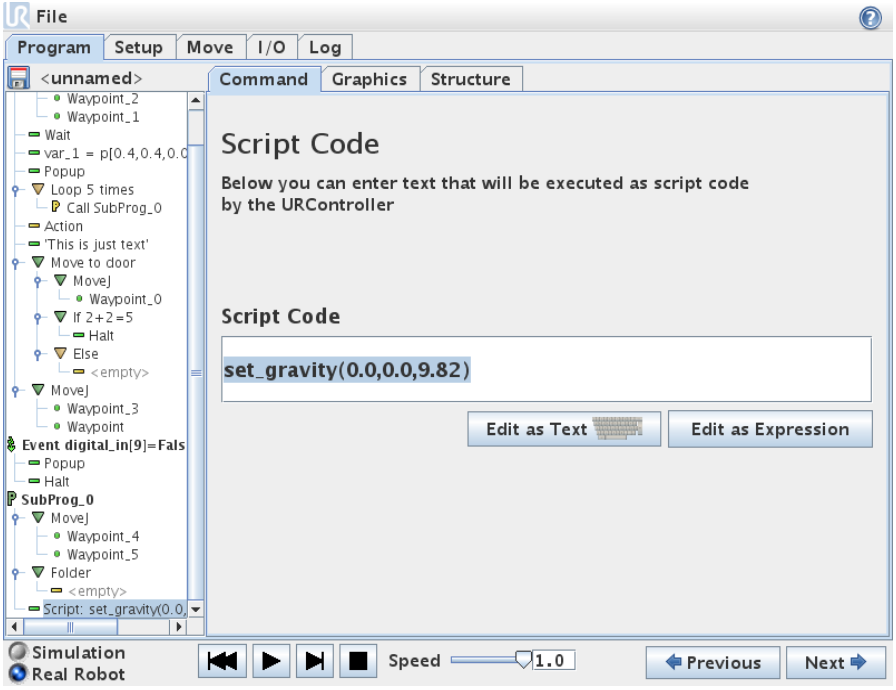



An "if..then..else" construction can make the robot change its behavior based on sensor inputs or variable values. Use the expression editor to describe the condition under which the robot should proceed to the sub-commands of this `If`. If the condition is evaluated to `True`, the lines inside this `If` are executed.

Each `If` can have several `ElseIf` and one `Else` command. These can be added using the buttons on the screen. An `ElseIf` command can be removed from the screen for that command.

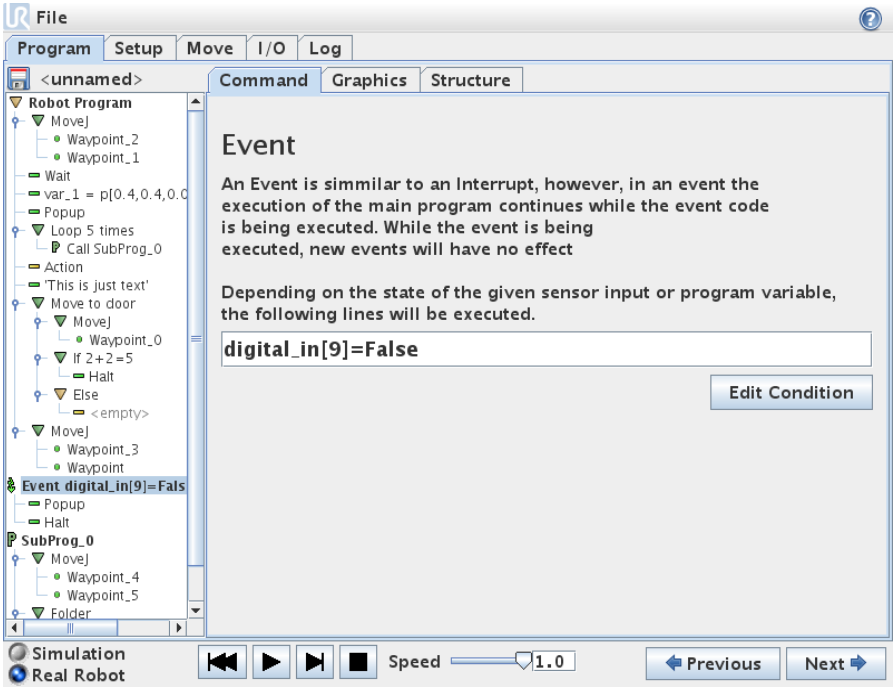
The open `Check Expression Continuously` allow the conditions of the `If` and `ElseIf` statements to be evaluated while the contained lines are executed. If a expression evaluates to *False* while inside the body of the `If`-part, the following `ElseIf` or `Else` statement will be reached.

### 3.4.18 Program → Command Tab, Script



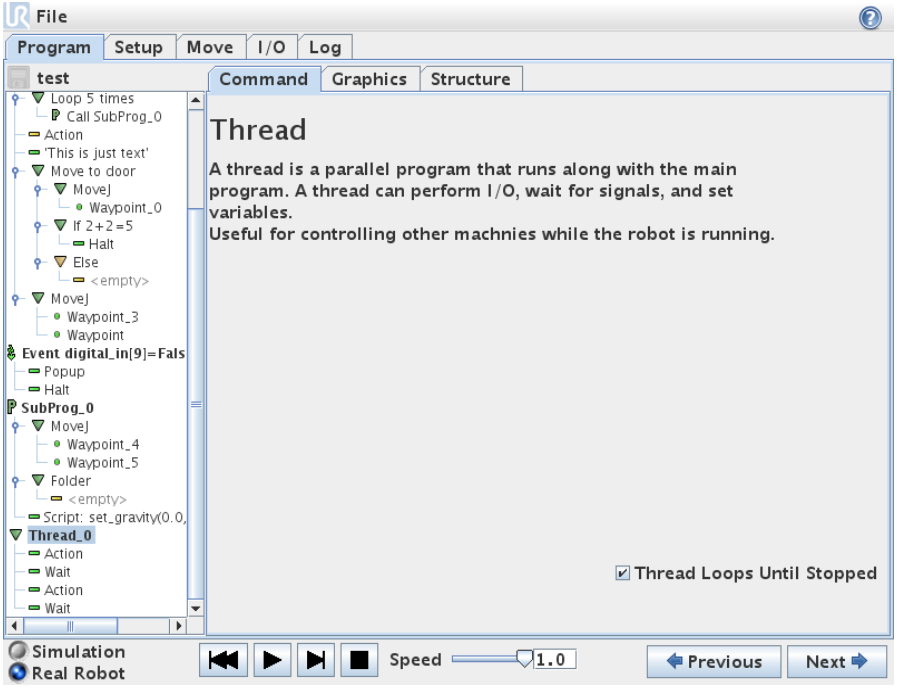

This command gives access to the underlying real time script language that is executed by the robot controller. It is intended for advanced users only.

### 3.4.19 Program → Command Tab, Event



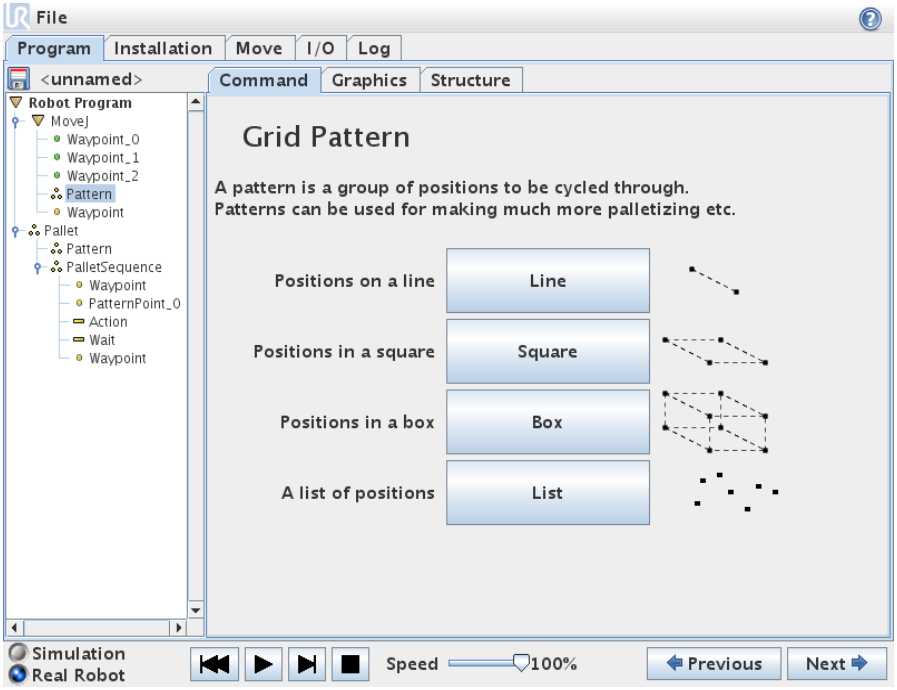

An event can be used to monitor an input signal, and perform some action or set a variable when that input signal goes high. For example, in the event that an output signal goes high, the event program can wait for 100ms and then set it back to low again. This can make the main program code a lot simpler in the case on an external machine triggering on a rising flank rather than a high input level.

### 3.4.20 Program → Command Tab, Thread




A thread is a parralel process to the robot program. A thread can be used to control an external machine independently of the robot arm. A thread can communicate with the robot program with variables and output signals.

### 3.4.21 Program → Command Tab, Pattern




The Pattern command can be used to cycle through positions in the robots program. The pattern command corresponds to one position at each execution.

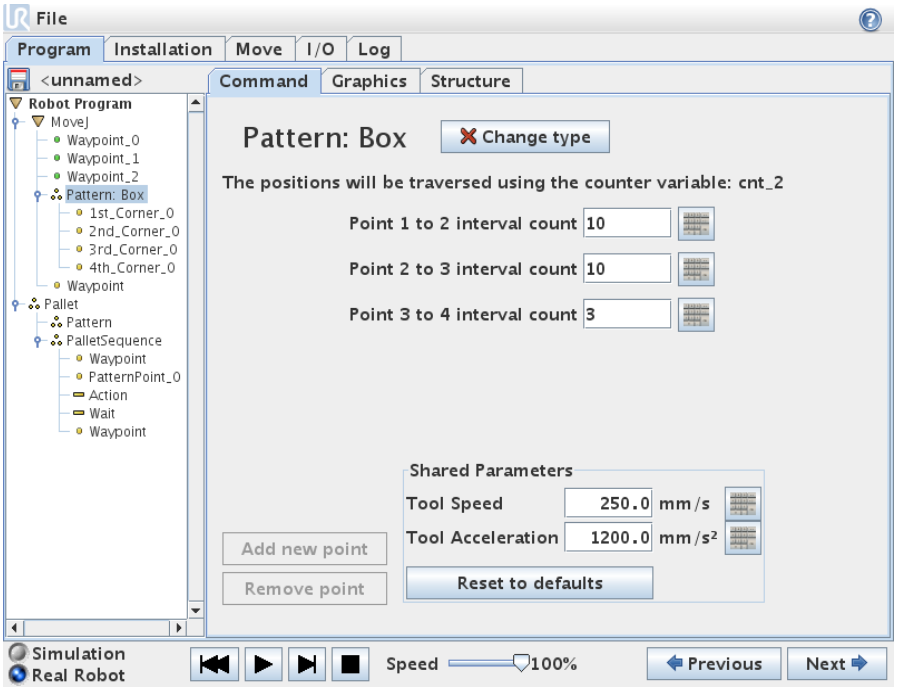
A pattern can be given as one of four types. The first three, "Line", "Square" or "Box" can be used for positions in a regular pattern. The regular patterns are

defined by a number of characteristic points, where the points define the edges of the pattern. For "Line" this is the two end points, for "Square" this is three of the four corner points, where as for "Box" this is four of the eight corner points. The programmer enters the number of positions along each of the edges of the pattern. The robot controller then calculates the individual pattern positions by proportionally adding the edge vectors together.

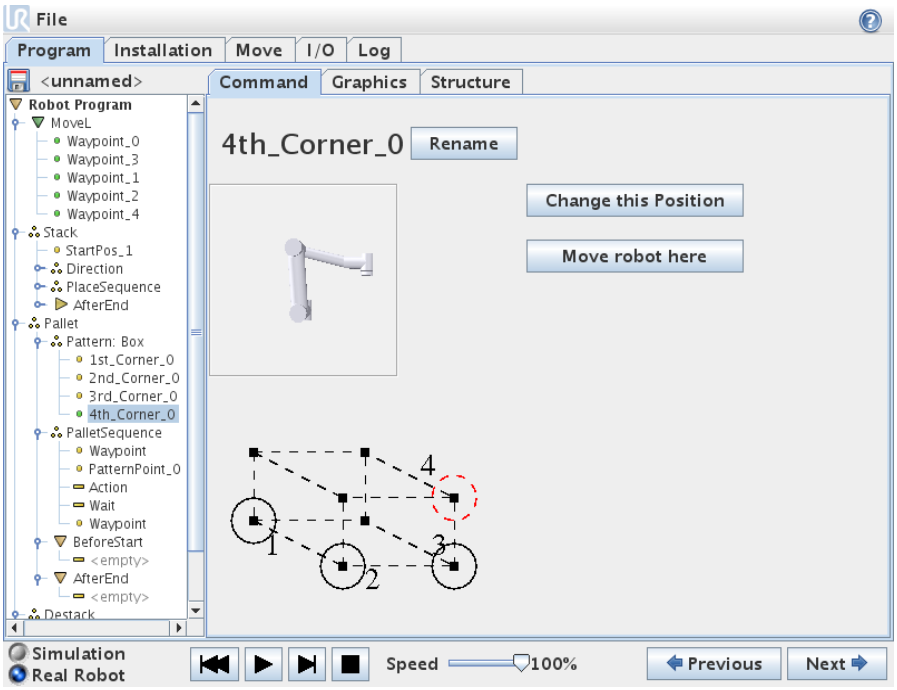
If the positions to be traversed do not fall in a regular pattern, the "List" option can be chosen, where a list of all the positions is provided by the programmer. This way any kind of arrangement of the positions can be realized.

### Defining the Pattern

When the "Box" pattern is selected, the screen changes to what is shown below.



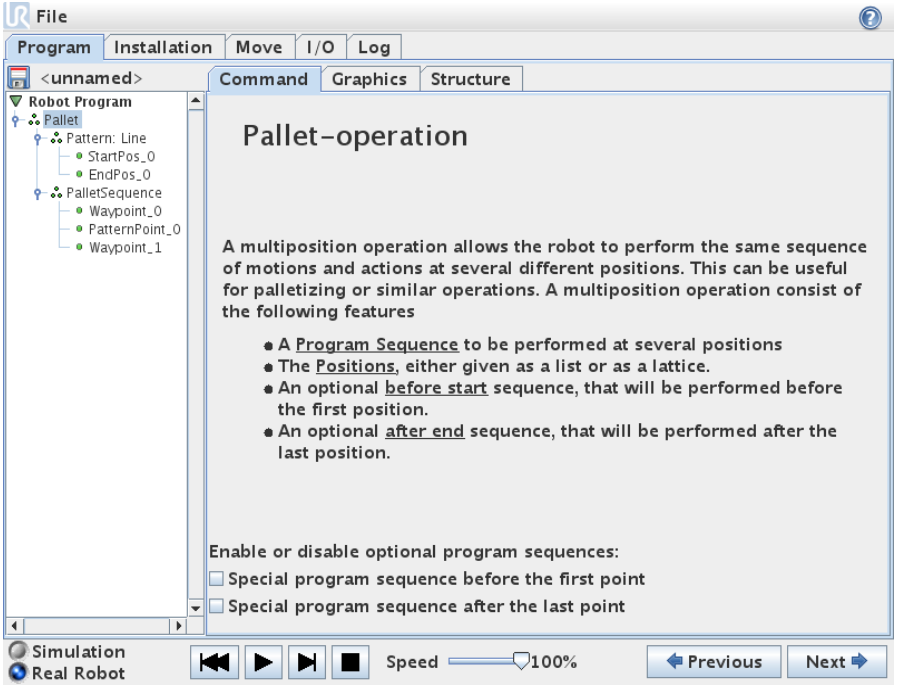

A "Box" pattern uses three vectors to define the side of the box. These three vectors are given as four points, where the first vector goes from point one to point two, the second vector goes from point two to point three, and the third vector goes from point three to point four. Each vector is divided by the interval count numbers. A specific position in the pattern is calculated by simply adding the interval vectors proportionally.




The "Line" and "Square" patterns work similarly.

A counter variable is used while traversing the positions of the pattern. The name of the variable can be seen on the `Pattern` command screen. The variable cycles through the numbers from $0$ to $X * Y * Z - 1$, the number of points in the pattern. This variable can be manipulated using assignments, and can be used in expressions.

### 3.4.22 Program → Command Tab, Pallet




A pallet operation can perform a sequence of motions in a set of places given as a pattern, as described in section 3.4.21. At each of the positions in the pattern, the sequence of motions will be run relative to the pattern position.

#### Programming a Pallet Operation

The steps to go through are as follows;

1. Define the pattern.
2. Make a "PalletSequence" for picking up/placing at each single point. The sequence describes what should be done at each pattern position.
3. Use the selector on the sequence command screen to define which of the waypoints in the sequence should correspond to the pattern positions.

#### Pallet Sequence/Anchorable Sequence

In an Pallet Sequence line, the motions of the robot are relative to the pallet position. The behavior of a sequence is such that the robot will be at the position specified by the pattern at the `Anchor Position/Pattern Point`. The remaining positions will all be moved to make this fit.

Do not use the `Move` command inside a sequence, as it will not be relative to the anchor position.

#### "BeforeStart"

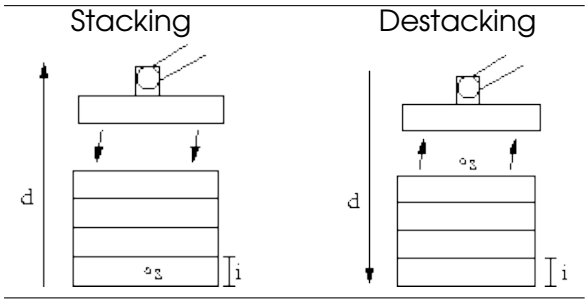
The optional `BeforeStart` sequence is run just before the operation starts. This can be used to wait for ready signals.

#### "AfterEnd"

The optional `AfterEnd` sequence is run when the operation is finished. This can be used to signal conveyor motion to start, preparing for the next pallet.

### 3.4.23 Program → Command Tab, Stack

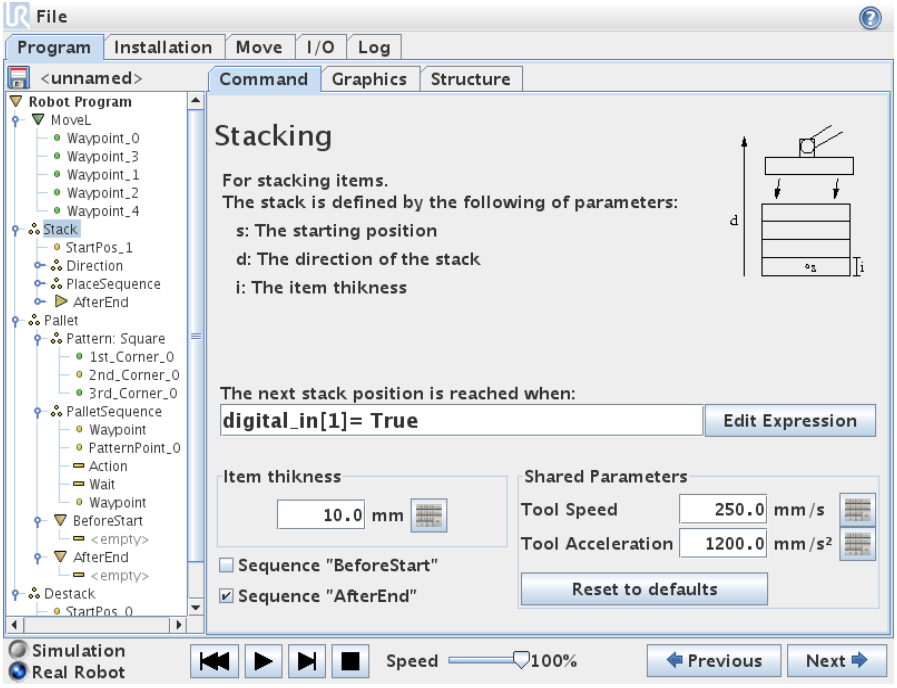
A stack is similar to a pallet with a "Line" pattern. However, stacking uses a sensor to determine when the correct position is reached to grab or drop an item. The sensor can be a push button switch, a pressure sensor or a capacitive sensor.




When programming a stacking operation, one must define $s$ the starting point, $d$ the stack direction and $i$ the thickness of the items in the stack.

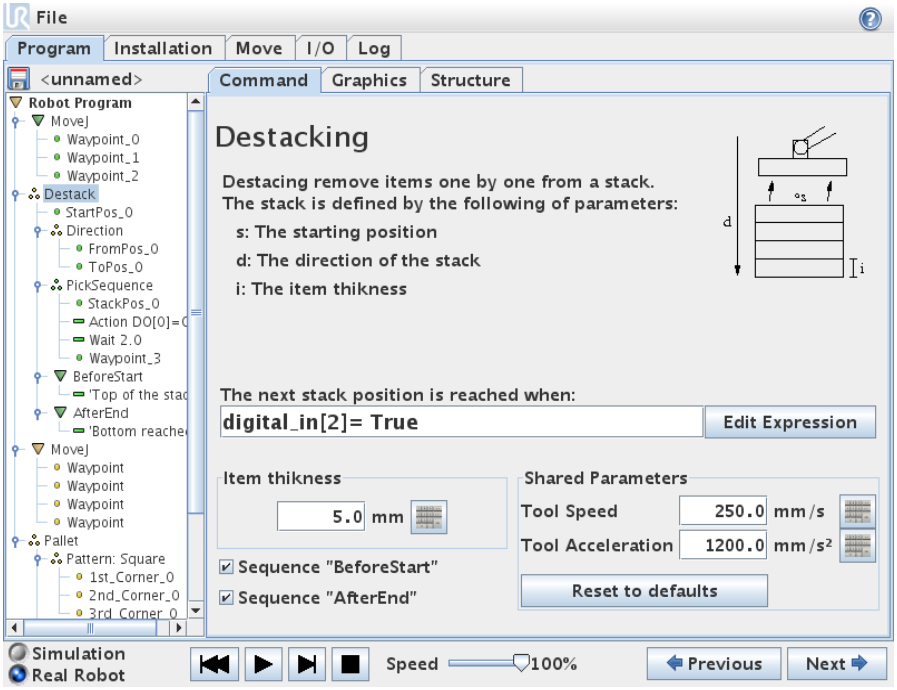
On top of this, one must define the condition for when the next stack position is reached, and a special program sequence that will be performed at each of the stack positions. Also speed and accelerations need to be given for the movement involved in the stack operation.

### Stacking




When stacking, the robot moves to the starting position, and then moves *opposite* the direction to search for the next stack position. When found, the robot remembers the position and performs the special sequence. The next time round, the robot starts the search from the remembered position incremented by the item thickness along the direction. The stacking is finished when the stack hight is more than some defined number, or when a sensor gives a signal.

### Destacking



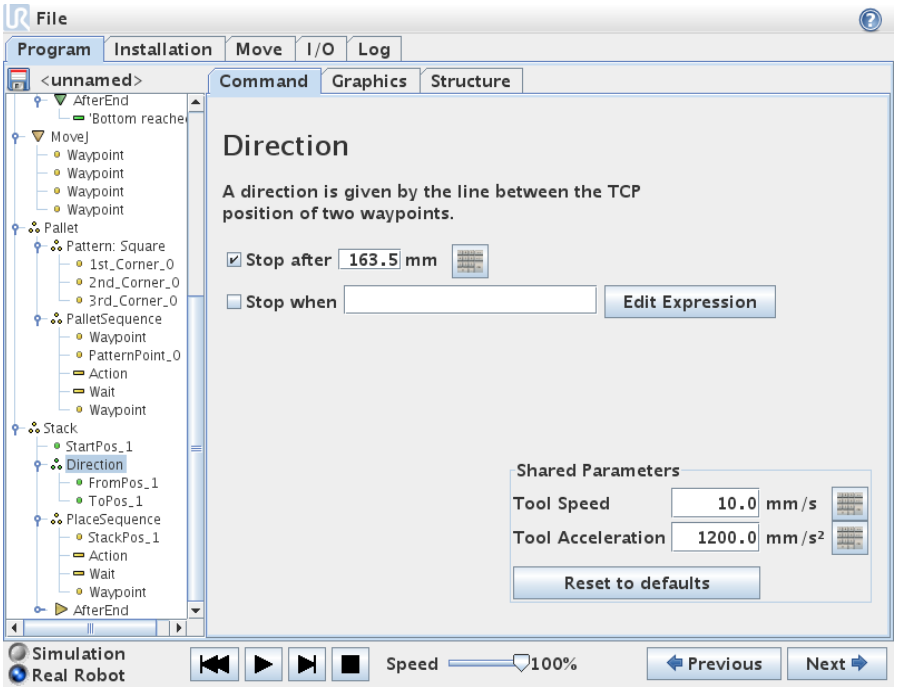

When destacking, the robot moves from the starting position in the given direction to search for the next item. When found, the robot remembers the position

and performs the special sequence. The next time round, the robot starts the search from the remembered position, incremented by the item thickness along the direction.

### Starting position

The starting position is where the stack operation starts. If the starting position is omitted, the stack starts at the robots current position.

### Direction




The direction is given by two points, and is calculated as the difference from the first points TCP to the second points TCP. [1]

### Next Stacking Position Expression

The robot moves along the direction vector while continuously evaluating whether the next stack position has been reached. When the expression is evaluated to `True` the special sequence is executed.

### "BeforeStart"

The optional `BeforeStart` sequence is run just before the operation starts. This can be used to wait for ready signals.

### "AfterEnd"

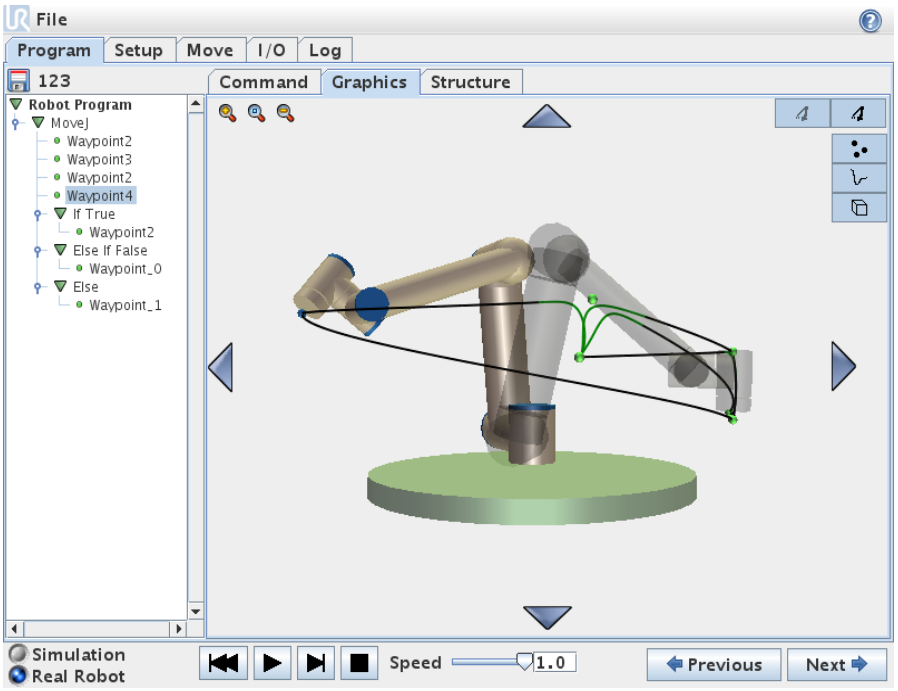
The optional `AfterEnd` sequence is run when the operation is finished. This can be used to signal conveyor motion to start, preparing for the next stack.

[1] A direction does not consider the orientations of the points.

#### Pick/Place Sequence

Like for the Pallet operation (3.4.22), a special program sequence is performed at each stack position.

### 3.4.24 Program → Graphics Tab

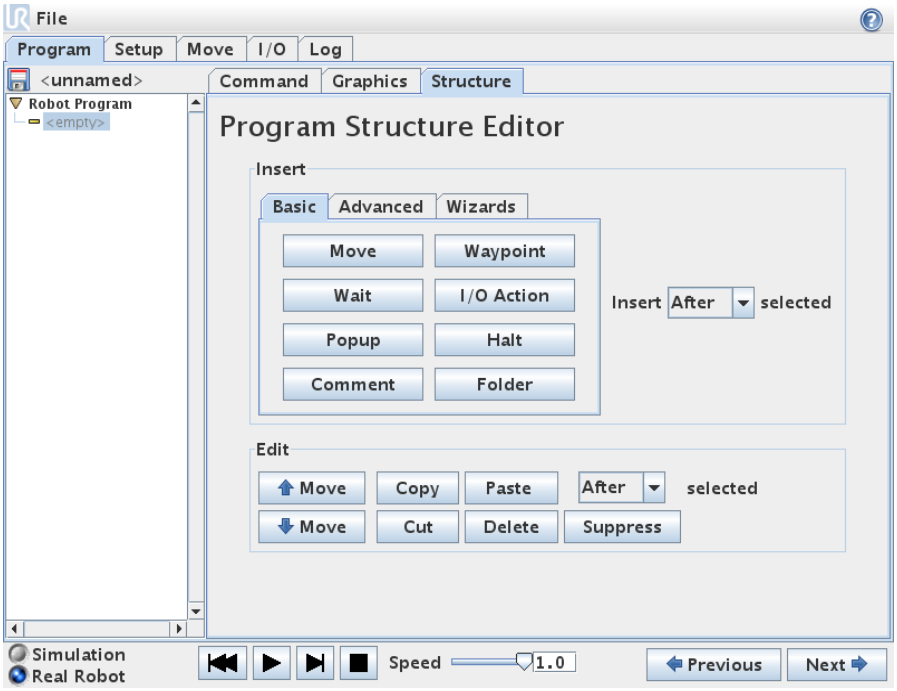



Graphical representation of the current robot program. The path of the TCP is shown in the 3D view, with motion segments in black, and blend segments (transitions between motion segments) shown in green. The green dots specify the positions of the TCP at each of the waypoints in the program. The 3D drawing of the robot shows the current position of the robot, and the "shadow" of the robot shows how the robot intends to reach the waypoint selected in the left hand side of the screen.

The 3D view can be zoomed and rotated to get a better view of the robot. The buttons in the top-right side of the screen can disable the various graphical components in the 3D view.

The motion segments shown depends on the selected program node. If a `Move` node is selected, the displayed path is the motion defined by that move. If a `Waypoint` node is selected, the display shows the following $\sim 10$ steps of movement.

### 3.4.25 Program → Structure Tab




The program structure tab gives an opportunity for inserting, moving, copying and removing the various types of commands.

To insert new commands, perform the following steps:

1) Select an existing program command.

2) Select whether the new command should be inserted above or below the selected command.

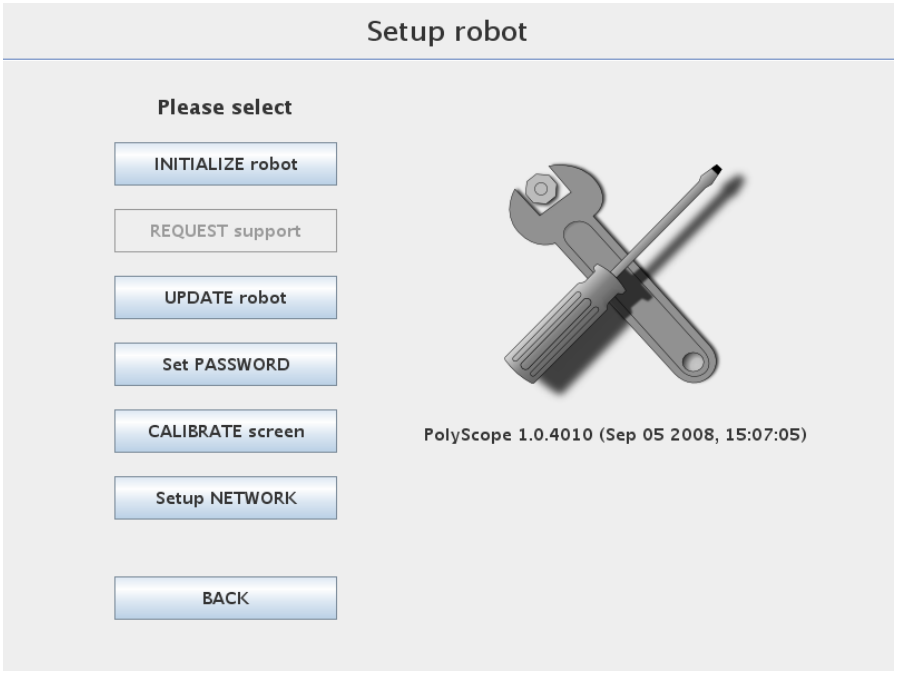
3) Press the button for the command type you wish to insert. For adjusting the details for the new command, go to the `Command` tab.

Commands can be moved/cloned/deleted using the buttons in the edit frame. If a command has sub-commands (a triangle next to the command) all sub-commands are also moved/cloned/deleted.

Not all commands fit at all places in a program. `Waypoints` must be under a Move command (not necessarily directly under). `ElseIf` and `Else` commands are required to be after an `If`. In general, moving `ElseIf` commands around can be messy. Variables must be assigned values before being used.

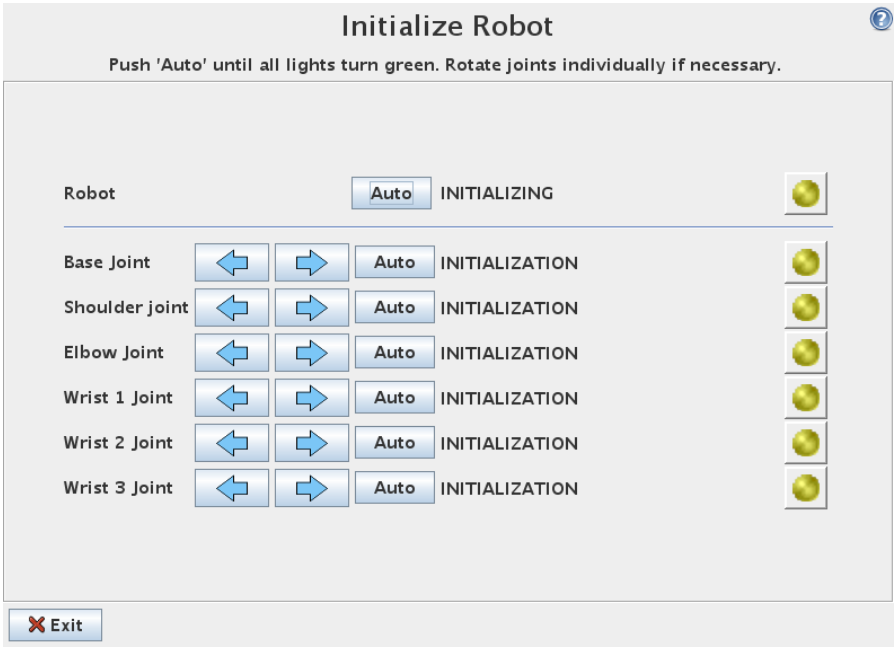
## 3.5 Setup

### 3.5.1 Setup Screen



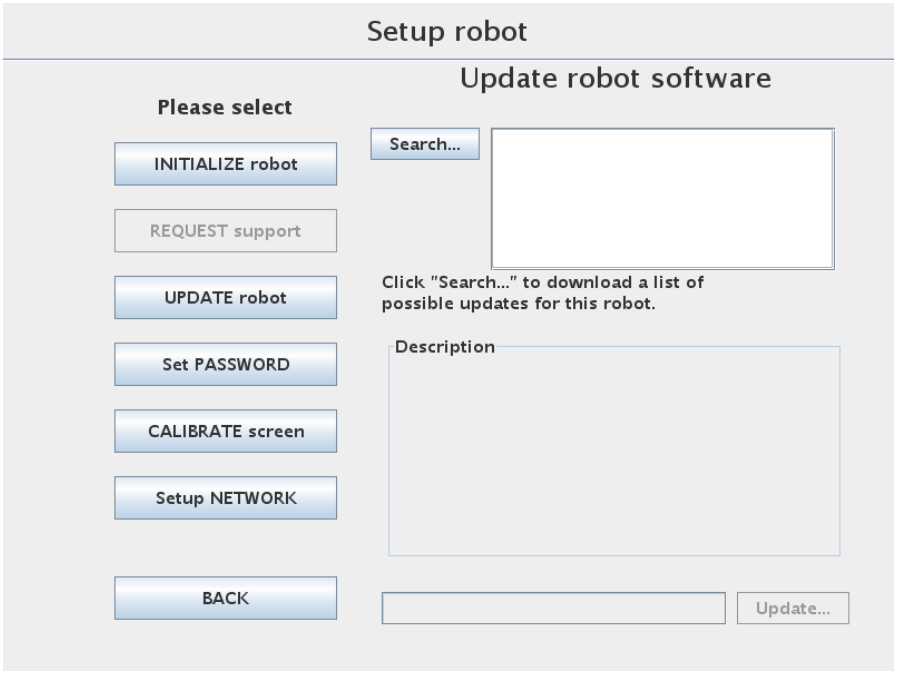

- **Initialize Robot** Goes to the initialization screen, see section 3.5.2.
- **Request Support** Opens a port in the robot that permits external access over the Internet.
- **Update** Upgrades the robot software to a newer version via the Internet, see section 3.5.3.
- **Set Password** Provides the facility to lock the programming part of the robot to people without a password, see section 3.5.4.
- **Calibrate Screen** Calibrates the "touch" of the touch screen, see section 3.5.5.
- **Setup Network** Opens the interface for setting up the Ethernet network for the robot, see section 3.5.6.
- **Back** Returns to the Welcome Screen.

### 3.5.2 Setup Screen → Initialize



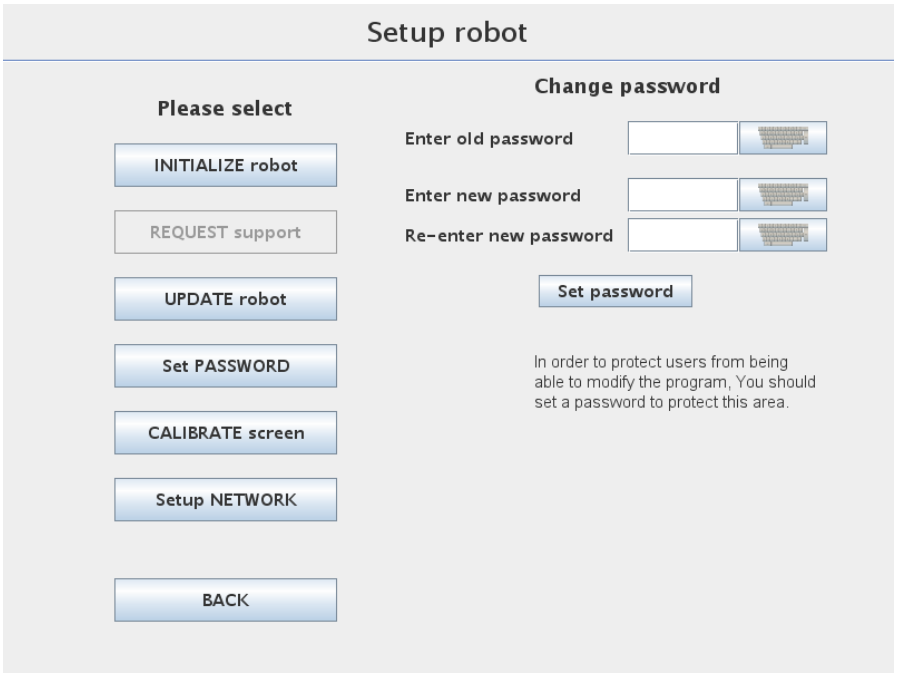

This screen is used when powering up the robot. Before the robot can operate normally, each joint needs to move a little (about 20°) to finds its exact position. The `Auto` button drives all joints until they are `OK`. The joints change drive direction when the button is released and pressed again.

### 3.5.3 Setup Screen → Update



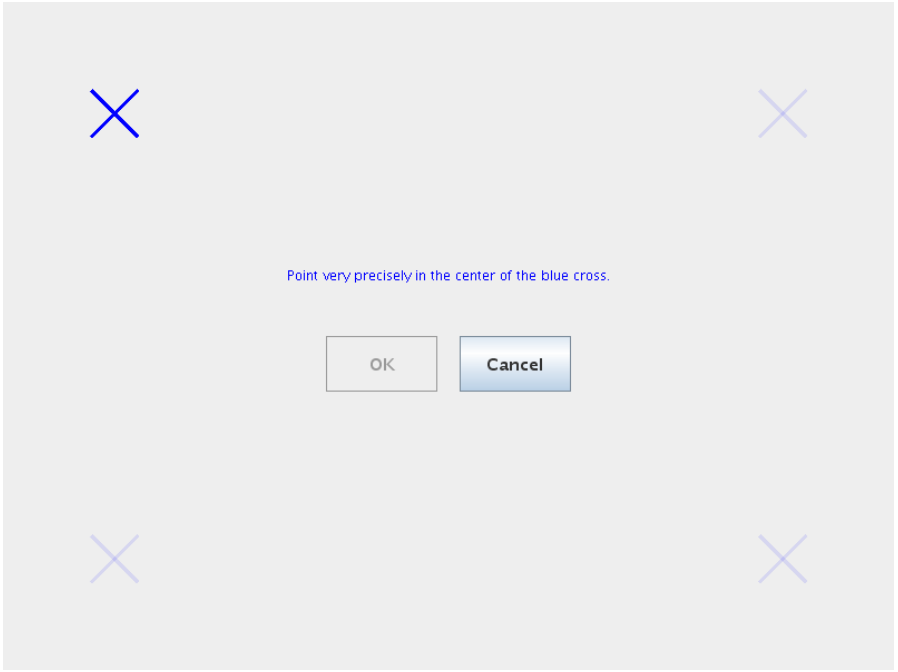

Provided the robot is attached to the Internet, new software can be downloaded.

### 3.5.4 Setup Screen → Password



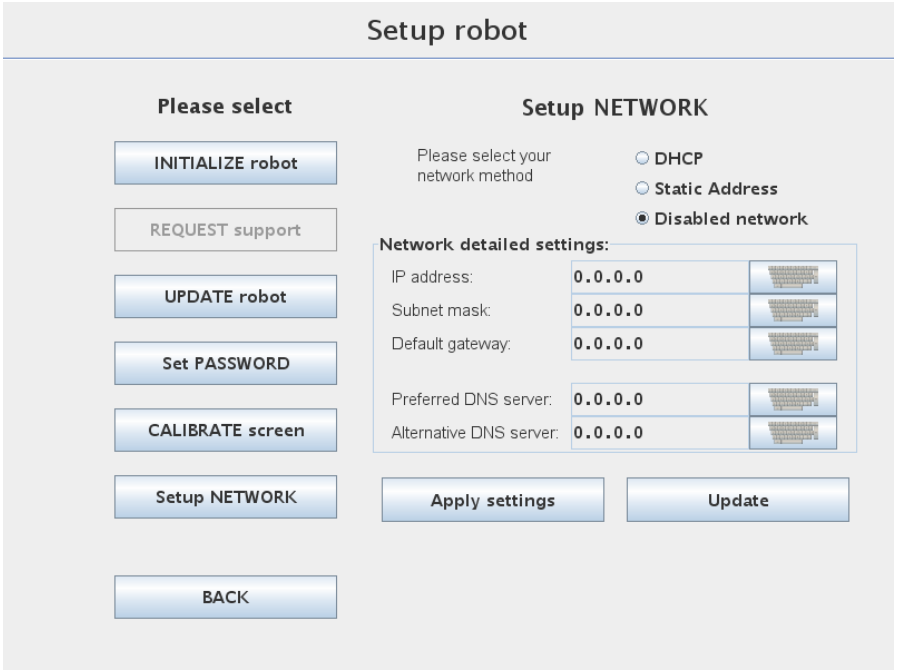

The programming part of the software can be locked using a password. When locked, programs can be loaded and run without the password, but a password is required to create or change programs.

### 3.5.5 Setup Screen → Calibrate Touch Screen




Calibrating the touch screen. Follow the on-screen instructions to calibrate the touch screen. Preferably use a pointed non-metallic object, such as a closed pen. Patience and care help achieve a better result.

### 3.5.6 Setup Screen → Network




Panel for setting up the Ethernet network. An Ethernet connection is not necessary for the basic robot functions, and is disabled by default.

# Chapter 4

# Warranties and Declarations

## 4.1 Warranty

### 4.1.1 Product Warranty

Without prejudice to any claim the user (customer) may have in relation to the dealer or retailer, the customer shall be granted a manufacturer's Warranty under the conditions set out below:

In the case of new devices and their components exhibiting defects resulting from manufacturing and/or material faults within 12 months of entry into service (maximum of 15 months from shipment), Universal Robots shall provide the necessary spare parts, while the user (customer) shall provide working hours to replace the spare parts, either replace the part with another part reflecting the current state of the art, or repair the said part. This Warranty shall be invalid if the device defect is attributable to improper treatment and/or failure to comply with information contained in the user guides. This Warranty shall not apply to or extend to services performed by the authorized dealer or the customer themselves (e.g. installation, configuration, software downloads). The purchase receipt, together with the date of purchase, shall be required as evidence for invoking the Warranty. Claims under the Warranty must be submitted within two months of the Warranty default becoming evident. Ownership of devices or components replaced by and returned to Universal Robots shall vest in Universal Robots. Any other claims resulting out of or in connection with the device shall be excluded from this Warranty. Nothing in this Warranty shall attempt to limit or exclude a Customer's Statutory Rights, nor the manufacturer's liability for death or personal injury resulting from its negligence. The duration of the Warranty shall not be extended by services rendered under the terms of the Warranty. Insofar as no Warranty default exists, Universal Robots reserves the right to charge the customer for replacement or repair. The above provisions do not imply a change in the burden of proof to the detriment of the customer.

In case of a device exhibiting defects, Universal Robots shall not cover any consequential damage or loss, such as loss of production or damage to other production equipment.

### 4.1.2 Disclaimer

Universal Robots continues to improve reliability and performance of its products, and therefore reserves the right to upgrade the product without prior warning. Universal Robots expects the contents of this manual to be precise and correct, but takes no responsibility for any errors or missing information.

## 4.2 Declaration of Incorporation

According to the machinery directive 2006/42/EC, the robot is considered a partly completed machine. The following subsections corresponds to and are in accordance with annex II of this directive.

### 4.2.1 Product manufacturer

| | |
|---|---|
| **Name** | Universal Robots ApS |
| **Address** | Svendborgvej 102<br>5260 Odense S<br>Denmark |
| **Phone number** | +45 8993 8989 |
| **E-mail address** | sales@universal-robots.com |
| **International VAT number** | DK29138060 |

### 4.2.2 Person Authorised to Compile the Technical Documentation

| | |
|---|---|
| **Name** | Lasse Kieffer |
| **Address** | Svendborgvej 102<br>5260 Odense S<br>Denmark |
| **Phone number** | +45 8993 8971 |
| **E-mail address** | kieffer@universal-robots.com |

### 4.2.3 Description and Identification of Product

The robot is intended for simple and safe handling tasks such as pick-and-place, machine loading/unloading, assembly and palletizing.

| | |
|---|---|
| **Generic denomination** | UR-6-85-5-A |
| **Function** | General purpose industrial robot |
| **Model** | UR-6-85-5-A |
| **Serial number of robot arm** | |
| **Serial number of control box** | |
| **Commercial name** | UR-6-85-5-A |

### 4.2.4 Essential Requirements

The individual robot installations have different safety requirements and the integrator is therefore responsible for all hazards which are not covered by the general design of the robot. However, the general design of the robot, including its interfaces meets all essential requirements listed in annex I of 2006/42/EC.

The technical documentation of the robot is in accordance with annex VII part B of 2006/42/EC.

| | |
|---|---|
| **Applied directives** | 2006/42/EC Machinery Directive<br>2004/108/EC EMC Directive<br>2002/95/EC RoHS Directive<br>2002/96/EC WEEE Directive |
| **Applied harmonized standards**<br>(Under applied directives) | IEC 61000-6-2 ED 2.0:2005<br>IEC 61000-6-4 ED 2.0:2006<br>EN 61000-6-2:2005<br>EN 61000-6-4:2007<br>EN ISO 13849-1:2008<br>EN ISO 10218-1:2008 (Partly)<br>EN ISO 13850:2008<br>EN ISO 14121-1:2007 |
| **Applied general standards**<br>(Not all standards are listed) | EN ISO 9409-1:2004 (Partly)<br>EN ISO 9283:1999 (Partly)<br>EN ISO 9787:2000 (Partly)<br>EN ISO 9946:2000 (Partly)<br>EN ISO 8373:1996 (Partly)<br>EN 60947-5-5/A1:2005<br>IEC 60947-5-5:1997/A1:2005<br>ISO/TR 14121-2:2007<br>EN 60529+A1:2002<br>EN ISO 1101:2006<br>EN 20286-1:1993<br>EN 20286-2:1993 |

Note that the low voltage directive is not listed. The machinery directive 2006/42/EC and the low voltage directives are primary directives. A product can only be covered by one primary directive and because the main hazards of the robot are due to mechanical movement and not electrical shock, it is covered by the machinery directive. However, the robot design meets all relevant requirements to electrical construction described in the low voltage directive 2006/95/EC.

Also note that the WEEE directive 2002/96/EC is listed because of the crossed-out wheeled bin symbol on the robot and the control box. Universal Robots registers all robot sales within Denmark to the national WEEE register of Denmark. Every distributor outside Denmark and within the EU must make their own registration to the WEEE register of the country in which their company is placed.

### 4.2.5 National Authority Contact Information

| | |
|---|---|
| **Authorised person** | Lasse Kieffer<br>+45 8993 8971<br>kieffer@universal-robots.com |
| **CTO** | Esben H. Østergaard<br>+45 8993 8974<br>esben@universal-robots.com |
| **CEO** | Enrico Krog Iversen<br>+45 8993 8973<br>eki@universal-robots.com |

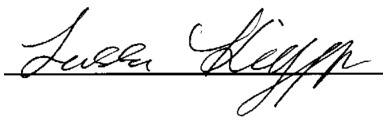
### 4.2.6 Important Notice

The robot may not be put into service until the machinery into which it is to be incorporated has been declared to be in conformity with the provisions of the Machinery Directive 2006/42/EC and with national implementing legislation.

### 4.2.7 Place and Date of the Declaration

| | |
|---|---|
| **Place** | Universal Robots ApS<br>Svendborgvej 102<br>5260 Odense S<br>Denmark |
| **Date** | 29. December 2009 |

### 4.2.8 Identity and Signature of the Empowered Person

| | |
|---|---|
| **Name** | Lasse Kieffer |
| **Address** | Svendborgvej 102<br>5260 Odense S<br>Denmark |
| **Phone number** | +45 8993 8971 |
| **E-mail address** | kieffer@universal-robots.com |
| **Signature** |  |

# Appendix A

# Safety Assessment

| Frequency, fr | |
|---|---|
| Daily | 5 |
| Weekly | 4 |
| Monthly | 3 |
| Annually | 2 |
| Less | 1 |

| Probability, pr | |
|---|---|
| Common | 5 |
| Likely | 4 |
| Possible | 3 |
| Rare | 2 |
| Negligible | 1 |

| Avoidance, av | |
|---|---|
| Impossible | 5 |
| Possible | 3 |
| Likely | 1 |

Table A.1: The three factors that need to be evaluated for each possible accident in a given robot application.

## A.1 CE certification of the Robot Installation

The robot is a CE certified machine[1]. In a given application, the robot is combined with a tool. The combination of the robot and the tool is a new machine. This new machine also has to be certified.

### A.1.1 Safety Requirements

The UR-6-85-5-A is a small and light industrial robot, with advanced motor control and surveillance of the robot's functions. Universal Robots has put much effort into making the robot as safe as possible. In Europe, the standard ISO 10218-1:2006 describes the requirements for a robot to operate without safety shielding. Most relevant here is §5.10.4, which states that the robot's tool should move at less than 250 mm/sec, that is 1 meter in 4 seconds. In addition, the maximum force exerted by the robot should be 150N, and the mechanical power should be less than 80W. The robot software has been written such that if the robot program moves the tool at less than 250mm/s, then these requirements are fulfilled, allowing the robot to drive without safety shielding. However, it is always important to be careful when around the robot.

### A.1.2 Scoring the Risk

For each potential accident in a given robot installation, the risk index of the accident has to be evaluated. To help evaluate this, this manual provides a safety evaluation form below. The form is based on the EN 14121-1 and -2 standards.

In a given robot installation, the risk assessment must be evaluated for each of the potential accidents. The risk assessment requires the evaluation of the Frequency, Probability and Avoidance for each of the potential accidents.

#### Frequency, fr

How often will there be a potential accident? For example, how often is a person within reach of the robot?

#### Probability, pr

How probable is it that the presence of a human will result in a dangerous situation?

[1] See section 4.2

| Seriousness | Class | | | | |
|---|---|---|---|---|---|
| | 3-4 | 5-7 | 8-10 | 11-13 | 14-15 |
| 4: Death, loss of eye or arm | D | E | F | G | H |
| 3: Permanent, loss of finger | C | D | E | F | G |
| 2: Reversible, Hospital | B | C | D | E | F |
| 1: Reversible, First Aid | A | B | C | D | E |

Table A.2: Risk Assessment for an accident

**Avoidance, av**

When an accident is about to happen, how likely is it that it can be avoided, for instance by jumping to the side or by pressing emergency stop.

### A.1.3 Risk Assessment

Estimate how serious the accident can be, and use this estimate in combination with the sum of values for frequency, probability and avoidance (in table A.1) to assess the risk class, using table A.2.

If the class is E,F,G or H, the safety around the robot installation should be improved. If the class is D, an effort should be made to improve the safety. If the class is A, B or C, ordinary care should be exercised around the robot.

### A.1.4 Example

Consider a robot installation, where a robot uses a suction disc to lift items from a machine to a box. The robot is not shielded, and programmed so it moves at less than $250mm/sec$. People often walk by the robot in a passage outside the robot's trajectory, but within reach of the robot.

There are two potential accidents in this installation.

1. An error in the robot's program can cause it to reach out into the passage.
2. A person moves close to the robot, for example to service the machine, without first stopping the robot.

A risk assessment needs to be performed for each accident, requiring that the *frequency*, *probability* and *avoidance* need to be evaluated for each accident.

**Example 1: Error in a robot program.** Errors in the robot program can happen whenever the robot has been reprogrammed, let's say daily, $fr = 5$. Errors in the robot program will only rarely cause it to move to the passage area, since the graphics on the programming screen shows the tool trajectory. This accident can only happen when a programmer has just programmed the robot and forgotten to check the robot's trajectory. Therefore, $pr = 2$. And since the robot is only moving at $250mm/s$, it is probable that the accident can be avoided, so $av = 3$. The sum, $fr + pr + av$, is $5 + 3 + 2 = 10$. Being hit by the robot can cause bruises that can be treated using first aid. Therefore, the lowest line of the risk assessment table is used, resulting in risk class $C$. The risk can be further reduced, for instance by blocking the passage of people during programming and testing of the robot.

**Example 2: Service without stopping robot.** A person forgetting to stop the robot when going to service the machine can happen weekly, $fr = 4$. It is likely that the presence of the person can result in a dangerous situation, $pr = 4$. It is also possible that the person realizes his mistake and avoids the accident, so $av = 3$. The sum is $11$. This accident will result in a bruise that can be treated by first aid, so the risk assessment class is $D$. Therefore, consideration should be given to what can be done to improve the safety of the installation, for example setting up warning signs, setting up a light breaker, or arranging the machine and robot so that the person won't be standing in the path of the robot.